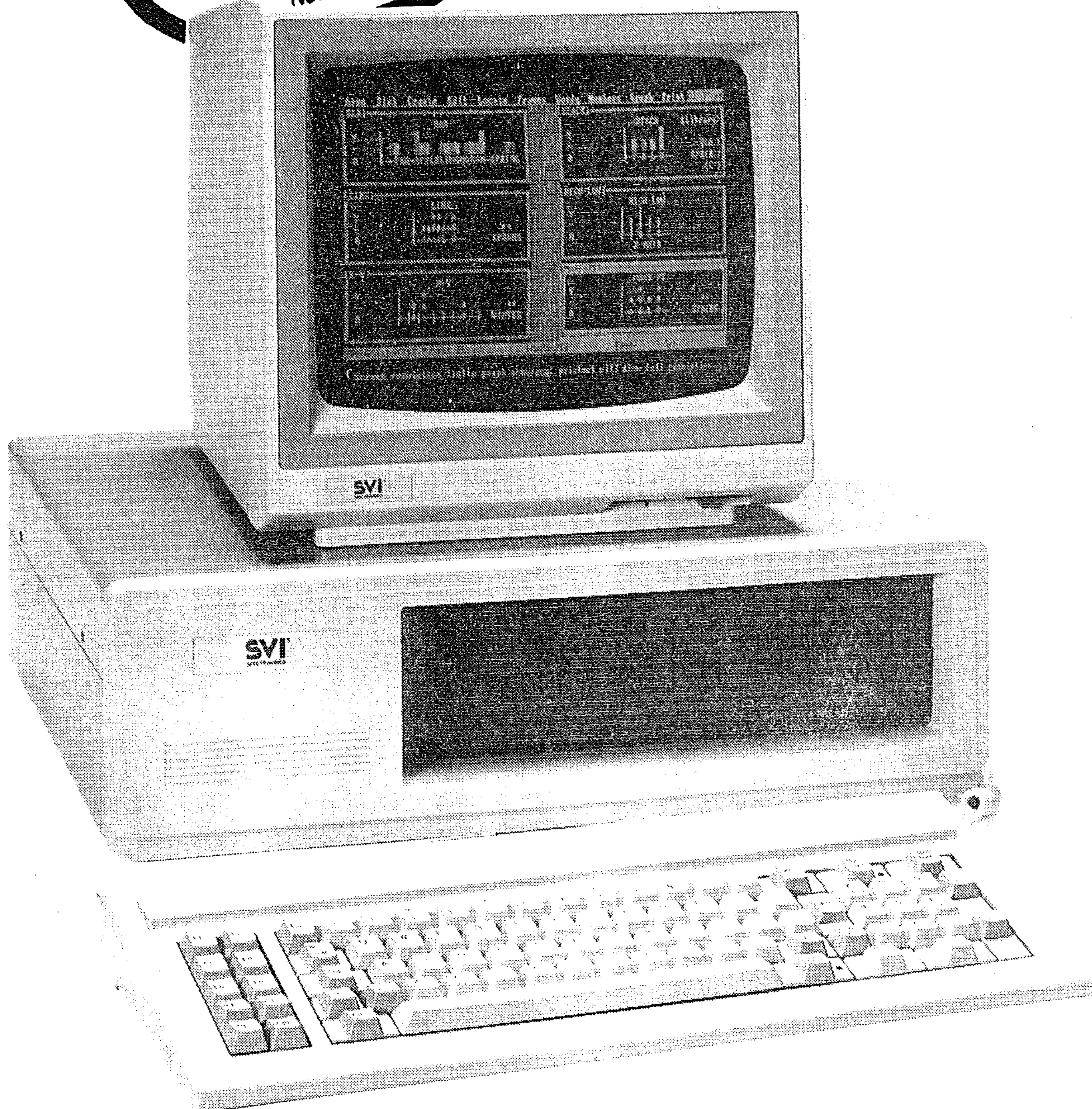
SYNTAX

SVI · MSX · PC

ÅRGANG 5 NR. 4 PRIS KR. 17

MAJ 1988

UDGIVET AF: MSX BRUGER KLUBBEN



- den ideelle PC-løsning

SVI-256SF
SVI-640FH
SVI-640FF
X'PRESS 16
Priser excl. moms - incl. MONOCHROM MONITOR

MEDLEMS RABAT 20 %

SVI
SPECTRAVIDEO

INCL. DOS 2.11
PÅ DANSK

Henvendelse:
OVERGÅRD ANDERSEN A/S
Dampfærgevej 32
2100 København Ø
Telefon 01 42 30 00

INDHOLDFORTEGNELSE

Artikler, annoncer eller programmer der ønskes optaget i bladet, sendes til Henrik Gilvad eller Preben Lund. Bånd og disk sendes retur.

Leder

I lang tid har producenter og forhandlere af programmer gået rundt og brokket sig over softwarepirater, men nu ser det ud til at lysne.

For nyligt blev den første store bøde givet til en pirat.

Pudsigt nok hørte piraten til en af de grupper der bliver ramt økonomisk af softwarepirater.

Piraten er (var?) forhandler af PC-maskiner og programmer, for at få sine egne PC'ere til at se mere attraktive ud end andres gav han en kopi af DSI-tekst med hver maskine. Piraten var iøvrigt kendt for at slå plat på Public Domaine programmer.

Prisen på PC-ere er for tiden ret lav, men man skal kigge sig grundigt for før man køber ind.

Mange steder køber man maskinen UDEN ramkredse, dette betyder at totalprisen bliver forhøjet ca 600 Kr. for hver 256 K Byte ram. Skal man have fyldt sin AT maskine op kan man risikerer at prisen bliver fordoblet. Prisen på ramkredse er skruet op p.g.a. for lille produktion.

Henrik G.

O P L Y S N I N G E R

Foreningen hedder: MSX BRUGERKLUBBEN
Medlemsskab koster 165 Kr. (årligt)
Salgsannoncer er for medlemmer gratis

Medlemskab opnås ved henvendelse til foreningens kasserer, eller ved indbetaling af kontingent på postgironr. 8 2 0 6 0 8 2

HUSK at opgive navn, adresse, postnr. og by, samt tlf.nr. og maskintype.

Bladet hedder : SYNTAX
Oplag : 800 eks.
Udkommer : 10 gange årligt (ikke jan og juli).
Tryk : Foto offset.
Forside udført af :John Mortensen
Løssalgspris : 17,00 Kr.
Udkommer næste gang : Juni

Formand : Henrik Gilvad
Solrød Byvej 41
2680 Solrød strand
Tlf. nr. 03 14 36 57

Kasserer : Preben Lund
Tømmerstrædet 19
2620 Albertslund
Tlf. nr. 02 64 76 26

Redaktionsgruppen består af formand, kasser og følgende :

Richard Foersom	Tlf.nr.	01 39 39 94
Henrik Larsen	-	01 17 48 23
Peter Knudsen	-	02 17 76 23
Erik Steen	-	02 96 09 37
Christian Noval	-	02 62 02 01
Søren Mortensen	-	01 69 77 40
Jannik Storm	-	01 65 74 59
Per Underlien	-	02 64 09 66
Kim Andersen	-	02 94 26 74
Søren Nielsen	-	02 52 85 93
Jonas Tromborg	-	02 65 28 98
FYN		
Jens Møller	-	09 94 22 75
JYLLAND		
Johnny B.	-	06 82 66 98
SVERIGE		
Jan Bojstrup	00946.	40-21 36 41
Joakim Tornhill	.	40-91 96 58
Magnus Balldin	.	40-15 01 98
NORGE		
Kjell Johansen	00947.	02-28 24 45

OBS. SIDSTE FRIST FOR STOF TIL NÆSTE NUMMER ER : 8.5.1988

PLANLÆGNING AF PROGRAMMER

Planlægning af programmer er en god ting at gøre INDEN man sætter sig til tastaturet.

Vi har for snart lang tid siden anvist metoder på, hvordan det kunne gøres, men da der er kommet mange nye medlemmer til klubben, og da det ikke skader lidt repetition (der kunne jo være ting man ikke vidste ind imellem) - så vil vi her beskæftige os lidt med planlægning.

Det er vigtigt, at man gør sig helt klart, hvilke egenskaber det program, som man skal igang med at opbygge skal have. Man må opstille nogle krav: Hvad skal der komme ud af programmet når det er kørt? Hvilke oplysninger har man brug for, for at få sine krav opfyldt?

Det kan bedst illustreres ved et eksempel:

Jeg vil gerne lave en telefontavle over mine venner og bekendte.
Jeg må derfor gøre mig klart, hvad der helt præsist skal stå på min telefontavle.
Selvfølgelig skal der stå telefonnumre. Der skal også stå hvis telefonnummer der er.
Men skal der også stå adresse? Det må jeg tage stilling til.

Nu har jeg allerede stillet nogle krav til, hvad jeg vil have ud af mit program. For at gøre det overskueligt kan jeg tegne det i e diagram (kaldes så Warner-Orr-diagram).

on-

Dette første led antyder, at det ikke er nok at vide, hvad programet skal ENDE med - men et sted skal man jo starte, og vi kan derfor bygge videre på vort diagram, samtidig med vi gør os klart hvorfra data'erne til programmet skal komme.

Det er vigtigt, at vi ved helt nøjagtigt, hvordan vores telefontavle skal se ud inden vi begynder at programmere. Det bedste er derfor, at lave et udkast - eller nedskrive på et stykke papir, hvilke elementer den skal indeholde.

Det kunne se således ud:

```
Telefontavle af 1.1.88

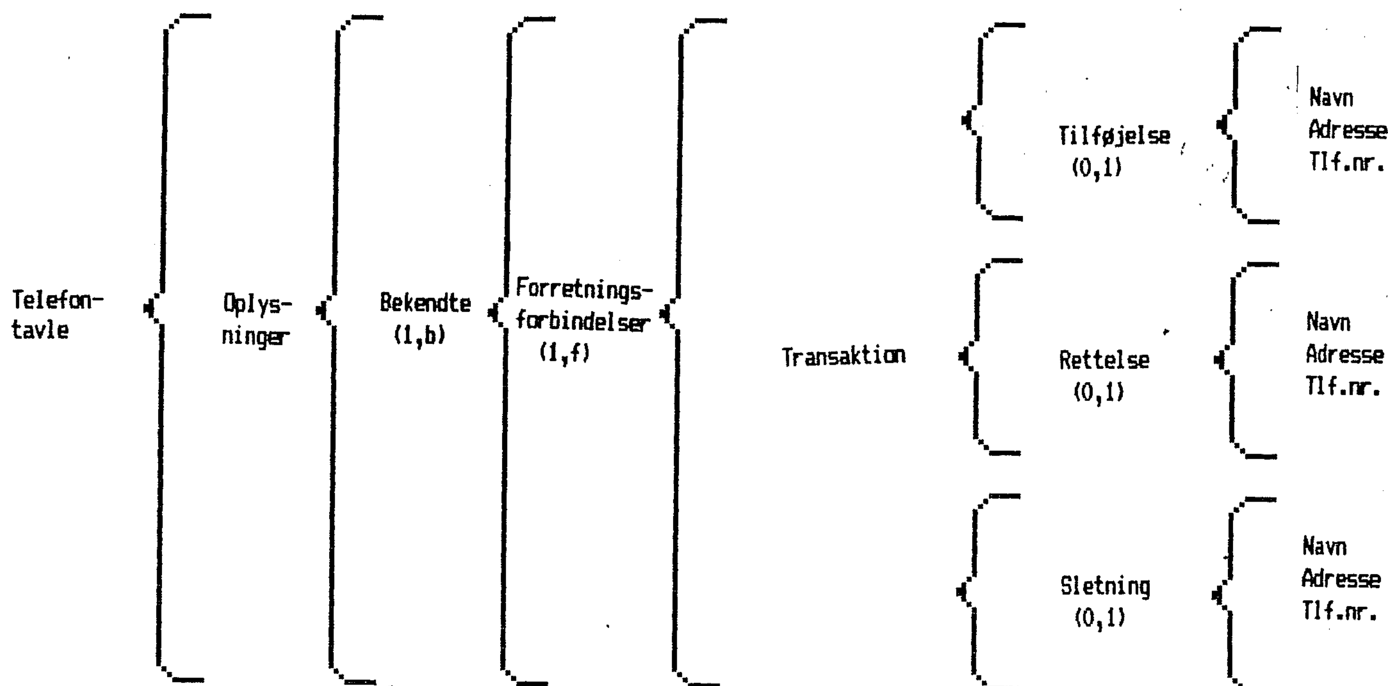
Navn          Adresse By ! Tlf.nr.
-------------------------!--------
                         !
Bekendt 1                !
Bekendt 2                !
    .                    !
    .                    !
Forretnings-             !
forbindelse 1            !
    .                    !
    .
```

En notation, som skal nævnes inden vi går videre er gentagelsesmarkeringen (1,a), hvor 1 står for at handlingen/forløbet skal gentages fra 1 til til a (variabel) antal gange.
Et andet symbol er "eller"-symbolet: ⊕ , som antyder, at man har et valg mellem flere muligheder.

Hvis vi skal fortsætte med vores telefontavle ser vi, at den har en opdeling i grupper. Samtidig ønsker vi, at kunne tilføje, rette og slette oplysninger på telefontavlen.

Hvordan gengiver vi dette i et diagram?

**** PLANLÆGNING AF PROGRAMMER ****

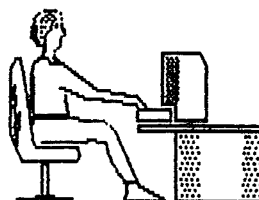


Med dette diagram har vi opdelt processen i enkeltdele. Tallene i paranteserne angiver, at der er tale om flere bekendte (1,b), og man skal samtidig forestille sig, at de findes i en rækkefølge, der begynder fra oven i klammen og slutter forneden.
På samme måde opdeles forretningsforbindelser naturligvis, som markeringen (1,f) betyder.

En telefontavle er ike meget værd, hvis ikke der er mulighed for, at foretage transaktioner. Disse transaktioner kan enten være tilføjelser, rettelser eller sletninger. I den sidste kolonne anføres den konkrete betegnelse af, hvilke variable det drejer sig om.

Det er af stor værdi, at konstruere en tegning/diagram af sit program inden man sætter sig til tastaturet - idet man herved får gjort sig klart, hvordan de forskellige processer skal forholde sig til hinanden indbyrdes, og hvilke oplysninger der er nødvendige til brug ved konstruktion af programmet. Samtidig giver det et overblik således, at man kan få et pænt struktureret program - og derved lettere undgå fejl. Endelig bliver programmet mere overskueligt, således at hvis man på et senere tidspunkt vil gøre eventuelle tilføjleser eller rettelser, kan dette ske meget nemt og smertefrit.

Det kan virke som spild af tid, at lave et diagram over forløbet af sit program, men den tid det tager, tjener sig ind senere ved lettere at kunne opbygge programmet og hurtigere lokalisere fejl.



**** PLANLÆGNING AF PROGRAMMER ****

Vi skal nu se på hvordan man opbygger et STRUKTURERET program helt fra bunden.

Til formålet har vi valgt spillet "at klunse".

For dem der ikke kender reglerne skal de her kort beskrives:

Der skal deltage mindst 2 spillere (hvor datamaskinen er den ene). Hver spiller får ved spillets begyndelse udleveret 3 pinde (tændstikker). Man skal nu lægge et antal pinde i sin knyttede hånd (kan være 0)

Derefter gælder det om, at gætte hvor mange pinde alle spillere har tilsammen. Den spiller, der gætter rigtigt, må lægge en pind til side. Det gælder om først at komme af med sine pinde.

Når vi har med computer at gøre, behøver man ikke, at bruge rigtige pinde, men kan nøjes med, at lade computeren holde styr på både sine egne og de deltagene spilleres pinde.

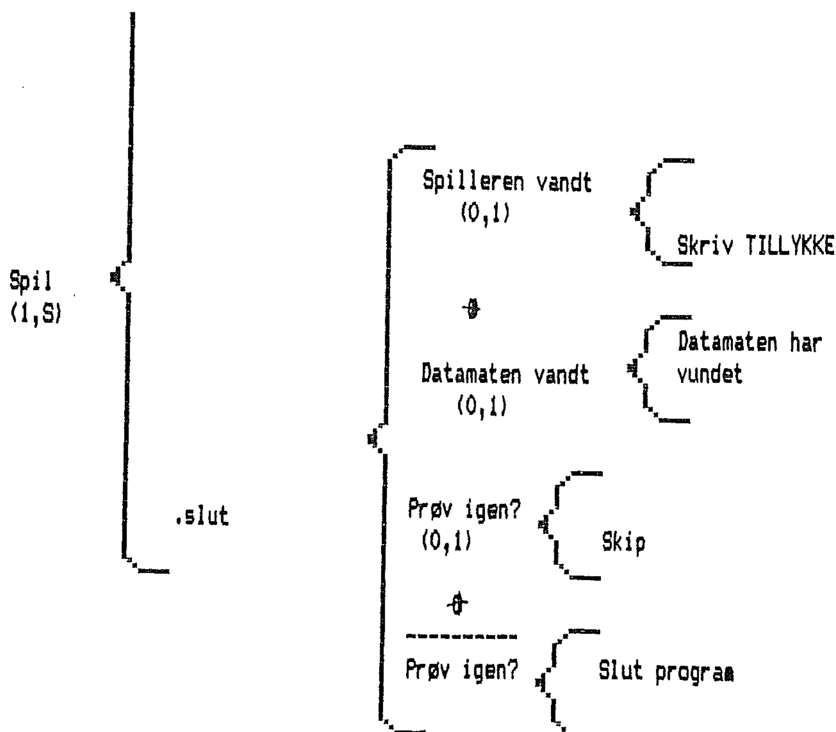
Vi begynder med, at lave en skitse til hvordan spillets struktur skal være:



Denne skitse viser, at der holdes styr på antal spil, hvis tur det er, samt antal spillere. I vort eksempel nøjes vi for forenklingens skyld med 2 spillere (computeren og 1 person)
Den sidste rubrik er "spil slut" rubrikken.

Her er angivet, hvordan afslutningen af et spil kan forløbe.

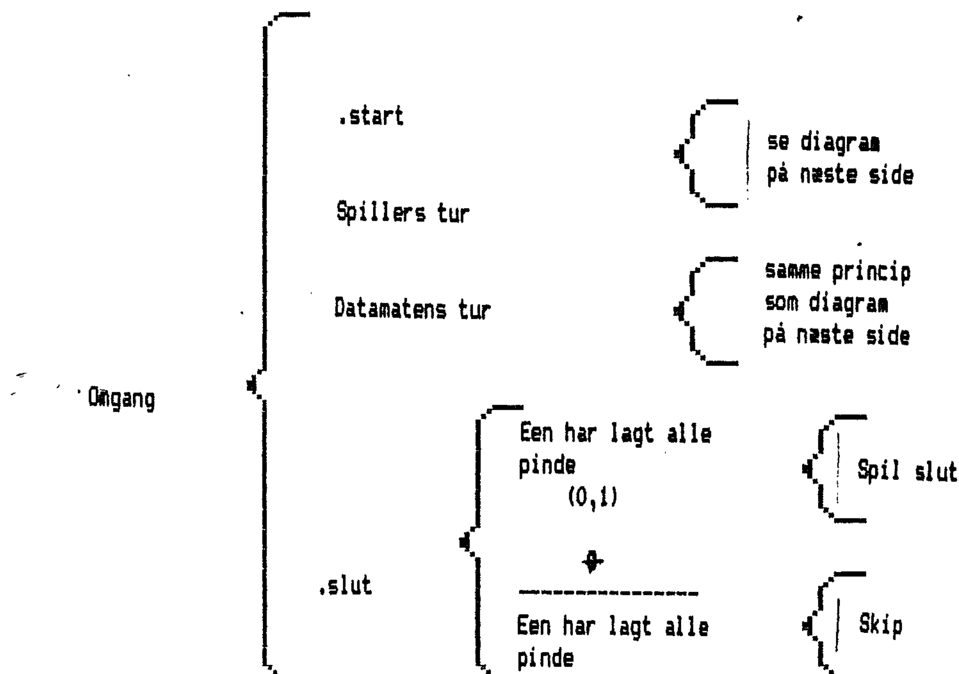
Ved omgang holder man styr på, hvor mange pinde hver spiller har - samt hvis tur det er.



**** PLANLÆGNING AF PROGRAMMER ****

Herefter følger 1. del af diagrammet til vores klunsespil. Til dette skal blot nævnes, at der forekommer steder, hvor den samme tekst står både over og under en vandret streg. Det skal forstås sådan, at teksten over stregen er positiv og teksten under stregen er negativ.

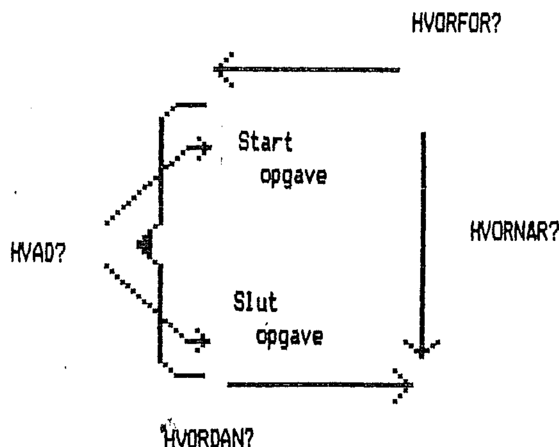
F. eks. betyder teksten: "een spiller har lagt alle pinde" - såfremt den står under en vandret streg: "INGEN spillere har lagt alle pinde".



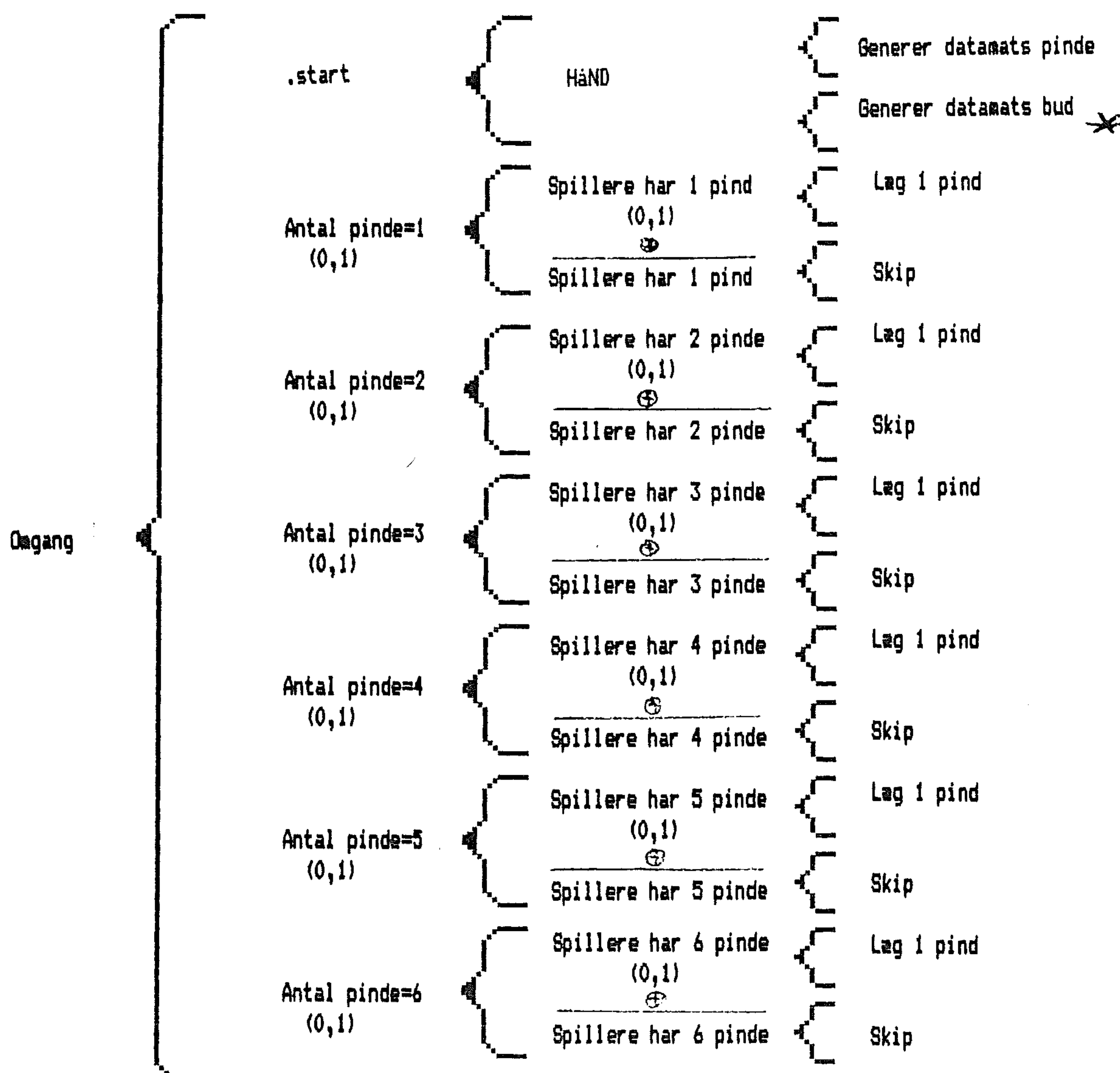
Strukturen i Warner-Orr diagrammer, kan bedst forklares med nedstående skitse.

Den giver svar på den logiske sammenhæng, på en mere overskuelig måde end flere siders forklaring ville gøre.

DIAGRAMMETS STRUKTUR



**** PLANLÆGNING AF PROGRAMMER ****



Til sidst vises programmet, som vort fine digram kan ende med. Jeg har med vilje ikke lavet grafik, selvom det jo er en oplagt opgave til dette program. Men det kan du jo selv lægge ind.

Bemærk, at programmet er struktureret opbygget. Det vil sige, at programforløbet styres fra en Styrings-sektion i programmet. Herfra bliver de relevante underafsnit kaldt. Fordelen herved er, at man kan genanvende programstumper, ved at kalde dem fra forskellige steder, samt at det bliver lettere at lokalisere fejl.

* DOG HVIS SPILLERS TUR:

- Indtast et tal
- Generer datamatens pinde

For at kunne overskue programmet vil det altid være en fordel, at give de enkelte dele navne i REM-sætninger, der fortæller hvad den pågældende del laver. Det er endvidere en stor fordel at sætte en REM-bemærkning efter alle steder hvor der er anvendt Gosub - da man så nemmere kan følge programforløbet.

En god regel er: Undgå at bruge GOTO. Hvorfor nu det? Jo for at undgå det man kalder Spahgetti-programmering. Det vil sige, programmer, der er fletet sammen i et spind af sætninger, der ikke er til at overskue.

**** PLANLÆGNING AF PROGRAMMER ****

Når der alligevel anvendes GOTO i nedstående program er det fordi BASIC ikke har DO...UNTIL kommandoen. Altså udfør et bestemt programforløb INDTIL en bestemt tilstand er opnået.
I denne situation er vi nødt til at anvende GOTO som "stopklods" med sætninger som If..xxxx then GOTO 999.

```
10 REM ***Klunsespil***
20 REM
30 REM antal spil - point
40 A=1
50 IFA>1 THEN GOSUB210:REM omgang
60 A$=""
70 IFA=1 THEN GOSUB120:REM spilstyring
80 REM omgang
90 GOSUB 210: REM omgang
100 REM spil slut
110 IF DA=0THEN GOSUB 790 ELSE GOSUB 86
0:REM vinder
120 REM ....spilstyring
130 PRINT;A;". Spil"
140 IF A>1 THEN  PRINT"Du har vundet";B
;" spil -jeg har vundet";C;" spil.":INP
UT "Vil du prøve igen? (J/N)";B$
150 PRINT
160 IF B$="N" OR B$= "n" THEN SLU=3 ELS
E GOTO90
170 IF SLU=3 THEN END
180 A$=""
190 RETURN
200 REM
210 REM omgang **************
220 CLS
230 DA=3
240 DU=3
250 IF A=1 THEN INPUT"Vil du starte? (J
/N)";A$
260 IF A$="n" OR A$="N" THEN TUR=1
270 IF A$="j" OR A$="J" THEN TUR=0
280 A$=""
290 GOSUB420:REM datamatens pinde
300 GOSUB550:REM dine pinde
310 FOR T=1TO2
320 IF TUR=1 THEN GOSUB 470 :REM dit bu
d
330 IF TUR=0 THEN GOSUB 600:IF T=1 THEN
 350
340 NEXT T
350 GOSUB650:REM datamatens pinde
360 OMG=PI+SP
370 IF GAET= OMG THEN DA=DA-1:IF DA>0TH
EN GOSUB 700 ELSE RETURN
380 IF DIT = OMG THEN DU=DU-1:IF DU>0 T
HEN GOSUB 750 ELSE RETURN
390 DIT=0
400 GAET=0
410 IF SLU=0 THEN GOTO290
420 REM  datamatens pinde *********
430 GEM=PI
440 PI=INT(RND(-TIME)*4)
450 IF PI>DA THEN GOTO 440
460 RETURN
470 REM   datamats bud ********
480 TUR=0
490 IALT=PI+DU
500 GAET=INT(RND(-TIME)*7)
510 IF GAET>IALT OR DIT=GAET THEN GOTO
500
520 CLS
530 PRINT"Datamatens bud:";GAET
540 RETURN
550 REM dine pinde ********
560 CLS
570  INPUT"Hvor mange pinde har du frem
me?";SP
580 IF SP>DU THEN PRINT"så mange pinde
har du ikke":GOTO 570
590 RETURN
600 REM **dit bud***************
610 TUR=1
620 INPUT"Indtast dit bud ";DIT
630 IF DIT=GAET THEN PRINT"Du må ikke b
yde det samme som datamaten":GOTO 620
640 RETURN
650 REM datamats pinde *******
660 PRINT"Datamaten havde";PI;"pind(e)
fremme"
670 FORX=1 TO 2000:NEXT
680 RETURN
690 REM resultat *********
700 CLS
710 PRINT"Datamaten lægger en pind, og
har nu";DA;"pind(e)"
720 PRINT""
730 FORX=1 TO 2000:NEXT
740 RETURN
750 REM resultat********
760 PRINT"Du lægger en pind, og har nu"
;DU;"pind(e)"
770 FORX=1 TO 2000:NEXT
780 RETURN
790 REM datamat vinder ********
800 CLS
810 PRINT"Datamaten har vundet!"
820 TUR=1
830 A=A+1
840 C=C+1
850 RETURN
860 REM  du vinder ********
870 CLS
880 PRINT"Du har vundet!"
890 TUR=0
900 A=A+1
910 B=B+1
920 RETURN
```

peter

SVENSKA SIDAN

Förra gången gav vi er ett längre programexempel som vi lovade att förkorta. Det som är nytt i denna nya version är en kopieringsrutin som kopierar den översta tredjedelen av skärmen till de två övriga tredjedelarna. Notera att detta gäller endast för pattern generatorn och colour table. Kopieringsrutinen ligger på raderna 70-80.

```
10 SCREEN 1
20 FOR I=0 TO 7
30 READ P¤,C¤
40 VPOKE I,VAL("&H"+P¤)
50 VPOKE &H2000+I,VAL("&H"+C¤)
60 NEXT I
70 OUT &H81,0 : OUT &H81,&H84
80 OUT &H81,&H9F : OUT &H81,&H83
90 FOR I=&H1800 TO &H1AFF
100 VPOKE I,0
110 NEXT I
120 GOTO 120
130 DATA 00,11,3C,A1,7E,A1,FF,61,FF
,61,7E,A1,3C,A1,00,11
```

Rad 20-60 ritar dels mönster och dels färg i SCREEN 1.
Rad 70-80 kopierinsrutin.
Rad 90-110 ritar ut hamburgarna på skärmen.
Detta var väl enkelt. Vi kan glädja dig med att det inte kommer att bli mycket svårare än detta.
Det vi kommer att gå igenom denna gång är att man kan ändra nametable figuren medan den finns på skärmen. Antag att du har ritat ut en hel del figurer med pattern generatorn och colour table. Antag vidare att du har fyllt skärmen med ruta noll av nametablerna. Om du nu börjar att ändra mönster och färg till denna ruta kommer ändringarna att återspeglas på nametablerna.
Följande programexempel illustrerar detta på följande vis: Det kommer att studsa en skiva (med ett A) på skärmen som roterar.

Denna skiva kommer att lämna ett spår efter sig med likadana skivor som också roterar. Här kommer nu programlistningen:

```
10 DEFINT A-Z
20 COLOR 15,1,1
30 X=16 : Y=12 : X1=1 : Y1=1
40 SCREEN 1
50 FOR I=0 TO 7
60 VPOKE &H2000+I,&HF1
70 VPOKE &H2800+I,&HF1
80 VPOKE &H3000+I,&HF1
90 NEXT I
100 VPOKE &H1800,255
110 VPOKE &H1900,255
120 VPOKE &H1A00,255
130 FOR I=0 TO 7
140 READ D¤
150 D=VAL("&H"+D¤)
160 VPOKE I,D
170 VPOKE I+&H800,D
180 VPOKE I+&H1000,D
190 NEXT I
200 RK=RK+1
210 IF RK=11 THEN RESTORE : RK=0
220 VPOKE &H1800+X+Y*32,0
230 IF X=31 OR X=0 THEN X1=-X1
240 IF Y=23 OR Y=0 THEN Y1=-Y1
250 X=X+X1 : Y=Y+Y1
260 GOTO 130
270 DATA 3C,7E,C3,DB,C3,DB,7E,3C
280 DATA 00,3C,7E,C3,DB,7E,3C,00
290 DATA 00,00,7E,E7,FF,7E,00,00
300 DATA 00,00,00,FF,FF,00,00,00
310 DATA 00,00,7E,FF,E7,7E,00,00
320 DATA 00,3C,7E,DB,C3,7E,3C,00
330 DATA 3C,7E,DB,C3,DB,C3,7E,3C
340 DATA 00,3C,7E,DB,C3,7E,3C,00
350 DATA 00,00,7E,FF,E7,7E,00,00
360 DATA 00,00,00,FF,FF,00,00,00
370 DATA 00,00,7E,E7,FF,7E,00,00
380 DATA 00,3C,7E,C3,DB,7E,3C,00
```

Nu kommer en liten förklaring.
Rad 30 är X och Y-led på skivan. X1 och Y1 är skivans rörelseriktning.
Rad 50-90 är färgsättning på skivan på alla tredjedelarna.
Rad 100-120 är till för att gömma mönstret och färgen till NAMETABLE-bitarna m.h.a en NAMETABLE-bit.
Rad 130-190 är det ställe där mönstret ändras.
Rad 200-210 kontrollerar så att man

börjar om med datasatserna efter att dessa tagit slut.
Rad 220 sätter ut NAMETABLE nummer noll.
Rad 230-240 byter riktning av skivan vid kollision av kant.

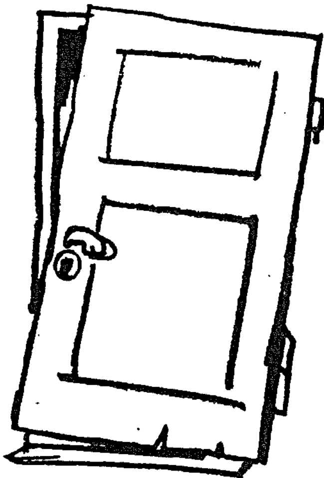
Som ni nog förstår ändrar programmet NAMETABLE-bit nummer nolls utseende på rad 130-190. Detta görs med hjälp av datasatserna vi har på raderna 270-380. Då vi ju hela tiden bara sätter ut NAMETABLE nummer noll på skärmen så ändras alltså alla NAMETABLE-bitar samtidigt.

Nästa gång kommer vi att gå igenom hur man använder VPOKE och VPEEK när det gäller tillverkandet av SPRITES.

Skulle det vara något ni inte förstått av det vi beskrivit hitintills så är det bara att ringa och fråga.

JOAKIM TORNHILL : 040-919658
MAGNUS BALLDIN : 040-150198

Nu säger vi tack och adjö för den här gången och må era ovänners kretsar kretsa.





MSX BIB

Nye programmer er der desværre ikke kommet til MSX-BIB siden sidst, men jeg fandt ud af at det fantastiske "LINESHOW" program kreeret af Richard de Foersom ikke var kommet med i samlingen. Som nogen måske kan huske det, er programmet udelukkende lavet til øjet, man kan ikke bruge det til noget som helst. Når man starter det begynder en slange af streger at bevæge sig frem og tilbage på skærmen. Og de bliver ved, og ved, og ved, og ved... Lige indtil man trykker på escape - så holder det op. Men så har man også fået en indre fred, der får ens aura til at føle sig som... Ja, køb programmet og du vil finde ud af hvad.

Ønsker du at komme i besiddelse af nogle programmer fra MSX-BIB er fremgangsmåden som altid:

Du sætter dig ned og finder ud af hvilke programmer du vil købe. Derefter slipper du joystikket for en stund, går hen på posthuset, fatter et giroindbetalingskort, og skriver programnavnene på bagsiden.

Priser:

15 min. bånd	- 25 dkr. stk.
5.25" disk	- 25 dkr. stk.
3.5" disk	- 35 dkr. stk.
programmer	- 10 dkr. stk.

****** MSX BIB ******

Et bånd kan rumme 8 programmer. En disk kan rumme hele biblioteket.

Eks: 8 programmer på 3.5" disk koster 8 * 10 + 35 = 115 dkr.

Programmerne bestilles hos:

Kim Andersen
Fundervej 32
2610 Rødovre
gironr. 7 50 87 00

Jeres egne programmer sendes til samme adresse. (der venter stadig en belønning i form af bånd eller disketter efter eget ønske).

Kim

Indhold i MSX-BIB:

- "**Lineshow**", meget flot demonstrationsprogram med bevægende linier.
- "**Kegle**", vælt den enlige bevægende kegle med din bowlingkugle.
- "**Figflyt**", kan flytte og dreje figurer du tegner ved at angive hjørnekoordinater. Regner nye ud.
- "**Sprite**", tegn din sprite og du får hvad der skal stå i datalinierne. Meget enkelt.
- "**Regression**", smart program der ud fra indtastede data (koordinater) tegner en tilpasset kurve.
- "**Tegne**", lille tegneprogram med de gængse kommandoer. Flot grafik.
- "**Snake Valley**", æd modstanderens hale inden han æder din. For 1 eller 2 spillere.
- "**Biorytmer**", afgør hvordan din fysiske, psykiske & intellektuelle tilstand er d.d.
- "**Chopper**", helikoptorspil med 12 baner, incl. designerprogram til at lave mange flere.
- "**Ormrun**", Charlottenlund Travbane med orme, flot grafik & lyd.
- "**Demo**", oplev hvilke grafiske muligheder din MSX indeholder.
- "**Amorti**", regner rente, afdrag og restgæld på din afbetaling ud.
- "**Tips**", indtast dine tips (også sys.) og prog. viser antal rigtige.
- "**The Maze**", lav en labyrint og få en anden til at slippe igennem.
- "**Life**", tegn et mønster af celler, og du ser dem udvikle sig.
- "**Sound Maker**", lav den rigtige lyd til dit sdpil på denne flotte mikserpult.
- "**Halv Tolv**", lige som i Monte Carlo, på med pokerfjæset.
- "**Danmark**", lær de danske byers beliggenhed, flot grafik.
- "**Oil**", bliv olie-sheik på no time, sjovt to-mands-spil.
- "**Puzzle**", sjovt puslespil på 16 brikker med bevæglige billeder.
- "**Skyd Nu**", skeetskydning som du kender det fra Ol, hurtig grafik.
- "**Pengo**", det iskolde gys, klassikeren fra spillehallerne.
- "**Søjle**", behandler tal og stiller dem op som kurver el. søjler.
- "**Eliza**", Psykologen hjælper dig med kærestesorgerne, på engelsk.
- "**Graf**", tegner en graf for den indtastede funktion, brugervenligt.
- "**Kartotek**", hold styr på dine cassette bånd og programmer.
- "**Moon Mission**", superlækkert grafik lander program, 5 baner.
- "**Frogrun**", få frøen sikkert over vejen og floden.
- "**Dump**", screendump prog. til din Epsonkompatible printer. 4 forstørrelser. 4 gråtoner.
- "**Dbase**", hurtig dbase hvor du kan sætte alt i system, indbygget udprintningsfacilitet.
- "**Screen Store**", kan gemme to skærmbilleder og lynhurtig få dem frem igen. Kun til 64 k maskiner.
- "**Ram Disc**", arbejd med to programmer på samme tid. Kun til 64 k maskiner.

SMÅTING TIL BEGYNDERE

Af JOHNNY B The GOOD

Husk kære venner at det lille tegn ' er lig med REM og det der stær ignoreres af computeren , du kan undlade at skrive REM sætningerne men husk linienummer + ' eller REM ellers går det nemt galt med et prg

Nu et lille prg der viser alle 16 farver på skærmen

```
10 SCREEN 0
20 PRINT"JOHNNY B MUSIK!"
30 FOR I=0 TO 255
40 VDP(7)=I
50 TIME=0
60 IF TIME<20 THEN GOTO 60
70 NEXT
80 SCREENO
```

Sprites er lidt svært dette lille program beregner sprite værdierne du laver ny sprite ved at ændre på 0 og 1 i data sætningerne

```
10 SCREEN 0
20 RESTORE
30 FOR J = 1 TO 8
40 READ N$
50 CHAR = 0
60 FOR I = 8 TO 1 STEP -1
70 X$ = MID$(N$,I,1):N=VAL(X$)
80 IFN=1 THEN CHAR=CHAR + 2 ^ (8-I)
90 NEXT
100 PRINT CHAR
110 NEXT
120 END
1000 DATA "10001000
1010 DATA "01010000
1020 DATA "00100000
1030 DATA "00111111
1040 DATA "00111111
1050 DATA "00100001
1060 DATA "00100001
1070 DATA "00100001
```

Værdierne sættes så ind det det prg herefter og spriten står på skærmen Prøv at slette REM i linie 80 og sæt derefter et Rem i linie 70

```
10 SCREEN 2,1
20 FOR I = 1 TO 8:READ N
30 A$ = A$ + CHR$(N)
40 NEXT:SPRITE$(1) = A$
50 PUT SPRITE 1,(100,100),1
60 GOTO 60
70 DATA 136,80,32,63,63,33,33,33
80 'DATA 56,56,16,56,84,16,40,68
```

Hvis vi ændrer programmet lidt så får vi bevægelse på skærmen husk REM det er det samme som før

```
10 SCREEN 2,1
20 FOR I = 1 TO 8:READ N
30 A$ = A$ + CHR$(N):NEXT
40 SPRITE$(1) = A$
50 FOR I = 0 TO 190
60 PUT SPRITE 1,(I,I),1
70 NEXT
80 GOTO 40
90 DATA 136,80,32,63,63,33,33,33
100 'DATA 56,56,16,56,84,16,40,68
```

Endnu en lille ændring så vi får lyd og bevægelse på vor sprite

```
10 SCREEN1,1
20 FOR I = 1 TO 8:READ N:
30 A$=A$ +CHR$(N):NEXT
35 SPRITE$(1) = A$
36 SPRITE$(2) = A$
40 X = 100:Y=50:X1=100:Y1=70
50 X3=2:Y3=2:X2=2:Y3=2
60 ONSPRITE GOSUB 180
80 PUT SPRITE 1,(X,Y),1
90 PUT SPRITE 2,(X,Y),1
100 IF RND(2)*6>3 THEN X2=-X2
110 IF RND(2)*4>3 THEN Y2=-Y2
120 IF RND(2)*6>3 THEN X3=-X3
130 IF RND(2)*4>3 THEN Y3=-Y3
140 X=X2+X:Y=Y2+Y
150 X1=X3+X1:Y1=Y3+Y1
```

```
160 SPRITE ON
170 GOTO 80
180 BEEP:BEEP
190 PRINT"OUCH!!"
200 SPRITE OFF
210 RETURN
220 DATA 136,80,32,63,63,33,33,33
230 'DATA 56,56,16,56,84,16,40,68
```

som du ser skiftede jeg til screen 1 du kan godt bruge screen 2 men da man da først skal lukke op skal du ændre flg linier

```
10 SCREEN2,1
190 PRESET(X2,Y2):PRINT#1,"OUCH!!"
```

+ indsætte flg linier

```
185 OPEN"grp:"AS#1
195 CLOSE
```

Prøv også at ændre alle demoerne til screen 3 idet alle sprites der bliver større - ligeså teksten

Ja nu kan du selv lege videre med sprites og hvis guderne og redaktøren vil vender jeg tilbage med mere til vore begyndere som jeg for ca 1 1/2 år siden selv hørte til
Venlig Hilsen Johnny B Silkeborg



BREV FRA ANDERS

I et brev vi har fået skriver Anders følgende:

Jeg har et problem som jeg håber i kan løse.
Mit problem består i at jeg får fejlmeddelelsen "String to long", og det er fuldstændigt korrekt, hvad computeren gør, idet at jeg bruger over 255 tegn i en streng. Jeg kan imidlertid ikke undgå det da jeg ikke kan slette indholdet i strengen, uden at ødelægge programmet.
Hvis man kan. Hvordan gør man og hvor meget kan man sætte af ?
Hvis man ikke kan det hvad (!! Censureret !! Red.) skal jeg så gøre ?
Programmet jeg er ved at lave er et slags orgel, det skulle gerne være muligt at man kunne få melodien gentaget, når man har 'Banket' den ind i computeren. Da der er noget der hedder volume, tempo og længde på en node når man hurtigt over 255 tegn i den streng, der skal gentage melodien. Hvis i kan løse dette problem kunne det jo være at i fik programmet til jeres blad og til MSX-BIB.

Med Venlig Hilsen:
Anders Juncker
"THE JUNKIE"

Svar:

Hej Anders.
Tak for brevet!
For at få en løsning på dit problem med det samme må du bladre et par sider frem for først kommer en gennemgang af MSX'ens måde at bruge rammen på.
ALLE adresser gælder kun for MSX mens principperne er næsten de samme på en SVI.

Da computerenn skal holde styr på en hel del har Microsoft været så flinke at give den en bid af ramen.

**** BREV FRA ANDERS ****

Denne del kaldes på engelsk for 'Workspace Area', direkte vil dette betyde 'Arbejds Område'.
Da os computerfreaks (Ja det blev jeg ihvertilfælde kaldt i skolen) godt kan lide at give ting tekniske navne vil jeg fremover kalde dette område for 'SYSTEM-VARIABLER'. 'SYSTEM-VARIABLERNE' fortæller hvor man finder de forskellige varibler, hvordan funktionstasterne ser ud samt meget meget meget meget mere.

Her har jeg tænkt mig at bladets udmærkede opsætter skal indsætte et skema over MSX-maskinernes RAM:

Pointernavn	Adr.(HEX)	Beskrivelse
		System variabler
HIMEM	FC4A	
		I/O Buffer 1
		FileControlBlok 1
		I/O Buffer 0
NULBUF	F862	
		FileControlBlok 0
		FCB-1 Pointer
		FCB-0 Pointer
FILTAB	F16A	
		er altid = 0
MEMSIZ	F672	
FRETOP	F69B	
		I dette område er alle $tringvaria-bler gemt.
STKTOP	F0A0	
		Z-80 stak
STREND	F6C6	
		DIMentionerede var gemmes her
ARYTAB	F6C4	
		!,#,% variabler + Streng beskrivelse (Incl. pointer)
VARTAB	F6C2	
		BASIC-Program
TXTTAB	F676	
		Denne byte ALTID=0
BOTTOM	FC48	

Skemaet viser rækkefølgen af de områder der gemmes i rammen fra adresse 8000h til FFFFh. Endnu et ord vil jeg nu udvide jeres horizonter med, nemlig et andet navn for et sådan skema:

* MEMORY MAP *

Nu er SYNTAX'en snart lige så horisont-udvidende som Anders And. (Men Anders And er i farver, SUK!)

Her en gennemgang af skemaet:
Til venstre står Pointernavnet og lige efter i hvilken adresse man kan læse værdien af denne pointer.
En Pointer optager 2 Bytes i systemvariablerne, for at regne indholdet af en pointer ud kan man skrive:

```
PRINT Peek(Adr)+Peek(Adr+1)*256
```

hvor Adr er pointeradressen.

Nederst ligger Basicprogrammet, det starter normalt på adresse 08001h men man kan godt snyde computeren til at tro noget andet ved at poke i TXTTAB før man loader et program ind. Dette triks bruger jeg selv på min 328 når jeg vil checke om mine programmer kan køre på en 318 (der jo kn har 16K ram)
Lad os betragte et basic program's måde at blive lagret i hukommelsen på:

<u>BASIC-PROGRAM</u>

```
11 PRINT "HEJ KURT"
14 END
```

<u>MEMORY</u>

```
8000      00
8001      11    Næste linie
8002      80    starter på Adr. 8011h

8003      0B    Denne linies nr.
8004      00    er nummer 11.

8004      91    Token for 'PRINT'
8005      20    " " = Mellemrum

8006-800F       teksten "HEJ KURT"

8010      00    0 betyder Slut på linie.
```

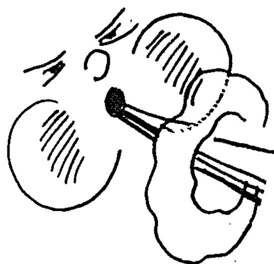
**** BREV FRA ANDERS ****

8011	17	Næste linie starter
8012	80	på adr. =8017h
8013	0E	Denne linie er nr.
8014	00	14 (0E i Hex)
8015	81	Tokens for END kommando
8016	00	Slut på linie
8017	00	og når 4 x 0 også slut
8018	00	på
8019	00	Program

Da computeren bruger 2 Byte's på at beskrive et linienr. kan et linienr have en værdi fra 0 til 65535.
Som i nok kan se fylder en kommando som PRINT kun en BYTE, denne byte er den interne kode (TOKEN) for PRINT, er du interesseret i TOKEN kan du finde dem i følgende gamle numre:

MSX-version, Dec.86
SVI-328 Nov 1986

Næste punkt i vores memory map er VARTAB:
Systemvariablen VARTAB fortæller hvor VAR-iabel TAB-ellen starter.
Denne tabel ligger lige efter der hvor BASIC programmet slutter, det er derfor alle variabler slettes når man udvider eller sletter en linie i sit program.
De eneste variabler der ikke ligger i vartab er de DIMentionerede, (Populært også kaldet indekserede variabler)
De indekserede variabler gemmes i en anden tabel der kaldes ARaYTABel, ARYTAB ligger efter VARTAB og afgrænses af STREND.
Ok, så trækker vi lige vejret et par linier.





Det var det, nu kan vi vist godt fortsætte.
Hver gang man tilføjer en ny variabel i sit program må computeren lyfte ARYTAB så der kan blive plads til en ekstra variabel i VARTAB. Denne proces tager faktisk tid, det kan derfor være en god ide at definere alle andre variabler end DIMerne først og så når dette er gjort kan man definere sine DIM. DIMer er meget hurtigere at bruge da de ligger helt fast i længde, ulempen er så bare at der ofte DIMes for stort hvilket spilder RAM. (Okay i behøver ikke hænge jer i det)
Her en kort beskrivelse af hvordan variabler gemmes i VARTAB.
Lad nu antage at vi i vores program fra før har en string variabel der hedder AB$ og som indeholder "SVI". Desuden har vi en almindelig var. Q% der er lig tallet 666.

	Adr.	Hex	Forklaring
VARTAB =	801A	03	Stringvariabel.
	801B	41	Den hedder A
AB$	801C	42	B
	801D	03	Den er 3 lang.
	801E	00	Den er lagret i
	801F	D5	adr. 0D500h
	....		
	8020	02	Heltalsvariabel.
	8021	3D	den hedder
Q%	8022	00	kun Q
	8023	9A	Dens værdi er
	8024	02	029ah = 666
	....		

**** BREV FRA ANDERS ****

ARYTAB = 8025
STREND = 8025 Der er ingen DIM variabler når ARYTAB = STREND

Som i kan se består en variabel i VARTAB af mindst 5 Byte's.

Byte	Beskrivelse.
1.	Beskrivelse af hvilken slags variabel det er.
2-3	Navnet på variablen.
4-5	Data: for Integer (%)
4-6	Data: $tring
4-7	: Single Precision.
4-11	: Double Precisionon

Forklaring for Byte 1.
Indholdet af byte 1 står for:

2 = Integer (Heltal %)
3 = String ($)
4 = Single presision (!)
8 = Doubel --!!-- (#)

Disse tal fortæller for talvar. egentligt hvor mange bytes ud over de 3 første. variablen bruger. i tabellen.

Forklaring af Byte 2-3.
Disse 2 Bytes indeholder Navnet.
I kan her se hvorfor computeren kun tager forbehold for de to første tegn i et variabelnavn.
HA$ og HANS$ vil altså være en og samme variabel.

De tre slags tal variabler :

Navn	Tegn	Længde
Integer	%	2
Single	!	4
Double	#	8

En % gemmes i 'almindeligt' format, d.v.s. som 2 Bytes der udregnes næsten lige som linienumrer. % er langt hurtigere for computeren at regne ud og med så det er altid en god ide at bruge dem i programmet hvor det er muligt.

! og # gemmes på en lidt speciel måde for at spare plads.
Et # tal kan indeholde hele 14 decimaler hvilket ikke sås på ret mange andre computere dengang MSX kom på markedet. Selv med 14 decimaler kan man nogen gange godt komme i klemme når man skal beskrive meget store eller meget små tal. Prøv at forestille dig et tal som 0.0000000000000123456, hvis dette tal skulle gemmes med 14 decimaler fik man ikke de 4 sidste med. Man har fundet på en smart måde at beskrive et sådan tal på, man tæller nemlig hvor mange man kan flytte kommaet mod højre før man har passeret det første rigtige tal. Gør vi det med vores tal kan vi se at vi kan flytte kommaet 13 pladserså vi får 1.23456. Vi mangler nu bare at fortælle hvor mange pladser kommaet er flyttet.
1.23456 E-13 beskriver dette på en kort og nem måde, når alle de kloge folk ude i verden skal bestemme små eller store tal gør de det faktisk på denne måde. De 6 tal vi har tilbage kan man skrive på BCD form så de kun fylder 3 Bytes.
12 34 56
Men man skal jo også vide om tallet er + eller - samt vi skal beskrive hvor mange gange vi har rykket kommaet. Dette kan gøres i en Byte

7	6	5	4	3	2	1	0
+/-	E-/E+						

Bit 7: +/- Tal
Bit 6: E- eller E
Bit 5-0: Exp.

Byte 7 er altså +/- bitten, et 1 vil betyde - og et 0 er +.
de resterende 7 bit (bit 0-6) kan indtage værdier fra -64 til 64 idet bit 6 giver polariteten.
Her er et eksempel til:
-12.3456 vil blive til C4 12 34 56.

**** BREV FRA ANDERS ****

Forklaring af Byte 4-?

Alle tal variablerne bliver gemt direkte i VARTAB mens en STRING variabel vil blive gemt et andet sted der bliver henvist til af Byte 5 & 6 i dens område i VARTAB.
$tring variabler gemmes nemlig i et område mellem STKTOP og MEMSIZ.
Størrelsen af dette område bliver defineret af Basic kommandoen CLEAR Skriver vi fx. CLEAR 300,&HA000 vil området være 300 Bytes langt.

Byte 4 fortæller længden, og her har vi roden til "JUNKIENS" problem da en BYTE jo kun kan have værdier fra 0 til 255.
DET ER IKKE MULIGT AT LAVE LÆNGERE TEKST-VARIABLER END 255 TEGN.

Området mellem STREND og STKTOP bruger Z-80 'en som stack til sig selv.
FILTAB pointeren peger på to pointere der igen peger på FCB. File Control Block'ene er 9 Bytes lang, betydningen kan ses i følgende tabel:

0	1	2	3	4	5	6	7	8
MOD	0	0	0	DEV	0	POS	0	PPS

File Control Blok

MOD : Buffer Mode
DEV : Device code
POS : 0-255,længde pointer
PPS : PrintPoSition,0-255

Byte 1,2,3,5,7 bruges ikke normalt.

MOD : For input = 01
For Output= 02
Random= 04
For Append= 08

DEV : CAS = FFh
LPT = FEh
CRT = FDh
GRP = FCh

POS : Fortæller hvor der nu skal læses i bufferen.

PPS : Fortæller hvor der skal skrives i bufferen.

Lige efter hver FCB følger en Buffer på 255 bytes, antallet af FCB bestemmes af kommandoen MAXFILES. Hvis man ønsker at åbne mange file samtidigt må man først angive hvor mange med MSXFILES, for hver åbnet fil bruges 255+9 bytes så man skal altså ikke flippe helt ud med maxfiles.

Da vi frygteligt gerne vil se Anders's program må vi jo hellere hjælpe ham lidt med nogen gode råd.

Man kan jo godt dele sine musik stykker op i mange variabler, men bedst vil være en DIMentioneret Strengvariabel.
Selv om man bruger en DIM kan man stadig kun have 255 tegn i, derfor skal man hele tiden checke om det er på tide at skifte til næste indeks.
Jeg har som eksempel på dette lavet et program hvor computeren husker hvad man har tastet og sørger for at der skiftes til næste indeks når den gamle er fyldt op.
Når man trykker <ESC> får man det hele printet ud på printer.
Efter programmet er der en lille gennemgang af programmet.

```
 10 CLEAR 3000
 20 DIM A$(10):S=0
 30 SCREEN 0:CLS
 98 ' Main program
100 T$=INPUT$(1)
105 'CHECK T$=<ESCAPE>
110 IF T$=CHR$(27) THEN 200
120 A$(S)=A$(S)+T$:?T$;
125 ' CHECK længde
130 IF LEN (A$(S)) > 200 THEN S=S+1
135 IF S=11 THEN 200
140 GOTO 100
200 ' PRINT ALLE TEGN.
210 FOR COUNTER=0 TO S
220 LPRINT A$(COUNTER);
230 NEXT
240 END
```

FORKLARING AF PROGRAM.

Linie	Forklaring
10	Reserverer plads for tekst.
20	DIMensionerer A$(0)-A$(10) Hvis man ikke DIM'er kan man kun bruge A$(0)-A$(9) eller fx. D(0)-D(9). Jeg DIM'er for at glæde min gamle EDB lærer der altid sagde at det er god programmering at gøre det. S sættes til 0
30	Stiller skærmen op.
100	Programmet venter til man taster 1 tegn, dette tegn bliver puttet ind i T$
110	Checker om <ESC> er tastet
120	Gemmer T$ i A$(s)
130	Checker om det er på tide at skifte variabel. Der skiftes ved at lægge 1 til s.
135	Checker om s bliver for stor da vi jo kun har dimmet til 11 (0-10)
140	Næste tegn
200	Udskrift til printer.
210	Start af FOR TO NEXT løkke.
220	PRINT tekst til printer.
230	Er jeg færdig, ellers så fortsæt.
240	END

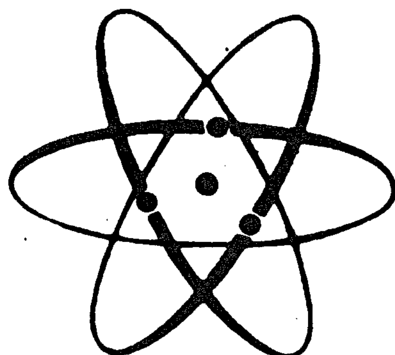
VARIABELDUMP-RUTINE.

Jeg har her lavet en lille rutine der udskriver alle normale variabler ved at læse i rammen ligesom computeren gør det. Du kan så selv prøve at ændre i variablerne i programmet og se hvordan det influerer.
Ved gennemkørslen kommer der pludselig noget GYF på en linie, det er p.g.a. linie 160 hvor jeg definerer en FN Z$. Denne funktion gemmes nemlig i samme område som variabler.
Dumprutinen printer ikke DIM. var.

```
10 ' Simpel VARIABEL DUMP rutine
20 ' H.G. 1 Dec. 1987
30 ' Rutinen dumper de almindelige
40 ' variabler ud på skærmen.
50 ' (Ikke DIM variabler !)
60 '
70 ' Def. Test variabler
80 P%=1
90 O!=4.123
100 I%=7
110 A$="Hello!"
120 R!=17
130 R#=17234.56789#
140 T#=1.71234E+61
150 '
160 DEF FNZ$(A)=CHR$(PEEK(A+1))+CHR
$(PEEK(A+2))
170 VARTAB = PEEK(&HF6C2) + PEEK(&H
F6C3)*256
180 A=VARTAB
190 ARYTAB = PEEK(&HF6C4) + PEEK(&H
F6C5)*256
200 C=PEEK(A)
210 PRINT
220 IF A>=AR THEN END
230 ON C GOTO ,250,290,370,,,,430
240 '
250 ' Integer
260 '
270 PRINT FNZ$(A);:PRINT"% =";:PRIN
TPEEK(A+3)+PEEK(A+4)*256;:A=A+5:GOT
O 200
280 '
290 ' String
300 '
310 PRINT FNZ$(A);:PRINT"$="CHR$(34
);
320 E=PEEK(A+4)+PEEK(A+5)*256
330 FOR D= E TO E+PEEK(A+3)-1
340 PRINTCHR$(PEEK(D));:NEXT
350 A=A+6:PRINTCHR$(34);:GOTO 200
360 '
370 ' Single precision
380 '
390 PRINT FNZ$(A);:PRINT"!=";
400 FOR Q=A+3 TO A+6
410 GOSUB 490
420 NEXT: A=Q : GOTO 200
430 ' Double precision
440 PRINT FNZ$(A);:PRINT"#=";
450 FOR Q=A+3 TO Q+7
460 GOSUB 490
470 NEXT : A=Q : GOTO 200
480 ' Udskrift i HEX
490 PRINT RIGHT$("0"+HEX$(PEEK(Q)),
2)" ";:RETURN
```

Henrik G.

**** PC BIB ****



Ja så er PCBIB på banen igen. Det er jo snart længe siden, siden men det skyldes at jeg har haft meget travlt, men fremover regner jeg med at der skulle være et PC BIB i hvert nummer.

PCBIB leverer software til PC ere. De penge der tjenes ind på PC BIB bruges til at købe mere software for, man kunne kalde det en evighedsmaskine. Prisen er 35 kr. pr. diskette portoen er betalt.

Der er utroligt mange disketter i PC BIB på nuværende tidspunkt, ca. 55 fyldte disketter. Da jeg desværre skal begrænse mig kan jeg ikke liste dem alle i bladet, men du kan sende en frankeret svarkuvert til mig så sender jeg dig en liste (adressen er sidst på siden) Du kan også kigge i nogle af de gamle numre og se hvad der er.

Alle programmer som er til PC ere og som bliver vist i bladet kommer i biblioteket, de havner på den såkaldte CLUBDISK. indtil vidrer er der ikke så meget på denne disk, det sidste nye er Kurts COLORTRIMMER, hvor nødigt jeg end vil rose en Kurt må jeg desværre indrømme at programmet er superlækkert, og meget brugervenligt, nu er det slut med at rode med svære farvekommandoer, med COLORTRIMMER mixer du lige den farve som du vil have, farven får derefter et nr. som bliver lagret i hukommelsen, og du kan fremover kalde denne farve frem.

Hvis du ligger inde med nogle spændende programer du selv har lavet eller som er PUBLIC DOMAIN (du skal være helt sikker !!) så tag og send dem til mig, du får dine disketter tilbage + samme antal nye disketter i belønning.

Ved bestilling af disketter, så gå ned til dit posthus, skriv dit navn og din adresse på girokortet, bag på girokortet skriver du hvilke disketter du ønsker. ca. 2 dage efter er dine disketter på vej til dig, der sendes disketter over hele Europa uden meromkostninger for modtageren, det skulle I tage at benytte jer af. Bestil allerede idag.

GIRO 5 9 9 3 3 5 0

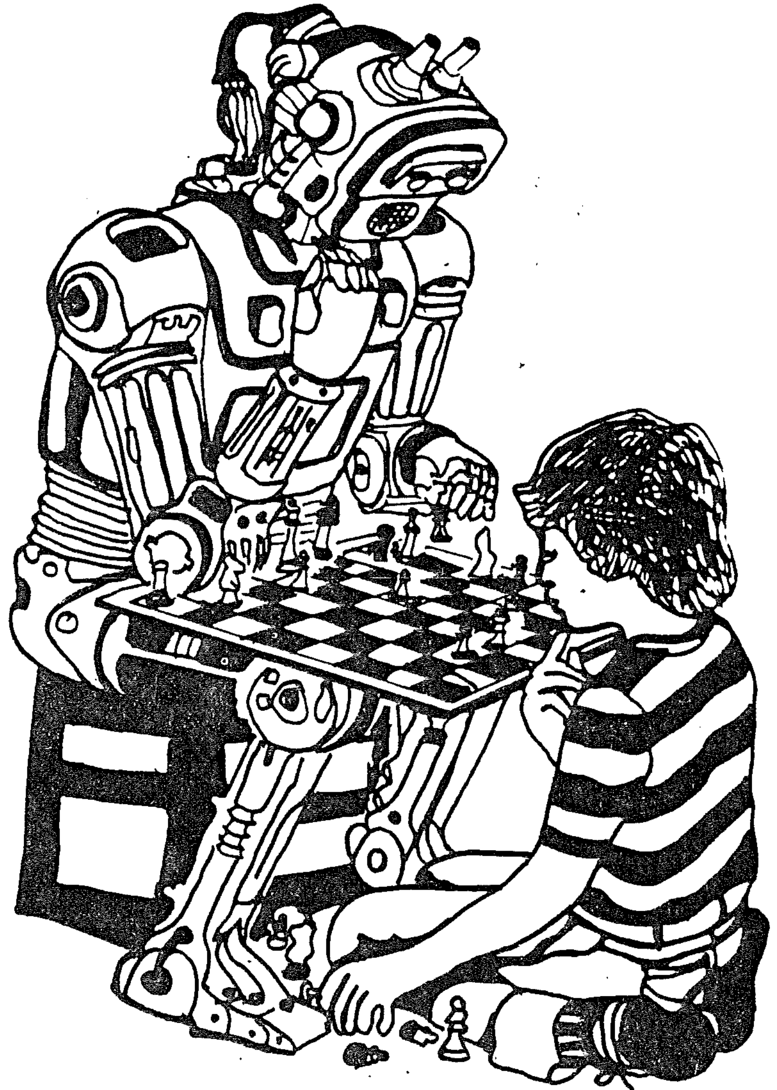
Henrik Larsen
Banevolden 44, lejl. 21
2500 Valby

Til sidst skal jeg lige nævne at jeg har fået kontakt med et software bibliotek i U.S.A som har ca. 5.000,00 disketter med Public Domain til PCere, jeg har allerede bestlit noget hjem så der er nok at glæde sig til. Jeg har også fået brev fra en i Norge som har noget Public Domain, det er meningen at vi skal udveksle noget software sammen. Så i har altså meget at se frem til, der skulle være nok programer til alle datafreaks i Norden.

Henrik

**** KØB/SALG & NYE MEDLEMMER ****

Køb & Salg



BILLIGE SPIL !!
Følgende spil er til salg :

Car Jamborie modul	50 kr.
Mole cule man	30 kr.
Reflex	50 kr.
Samlet køb	120 kr.

Anders Jørgensen
Torstedvej 40
6980 Tim
Jylland
tlf. 07 33 34 27

INDMELDELSESBLANKET

FORNAVN: ______________________________

EFTERNAVN: ______________________________

ADRESSE: ______________________________

TLFNR.: ______________________________

POSTNR. +BY: ______________________________

ALDER: ______________________________

MASKINE: ______________________________

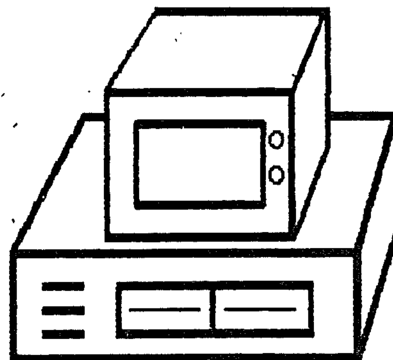
DIV. UDSTYR: ______________________________

INTERESSE OMRADE: ______________________________

**** KØB/SALG & NYE MEDLEMMER ****

TILBUD

SONY MSX



1 Stk. Sony Hit Bit HB-75P
1 Stk. Sony Diskdrive HBD-50
1 Stk. Joystick Quikshot II turbo
Diverse Spill og Nytte-
programmer.
Orginale Esker.
Diverse bøker.
Over 1. års garanti.
Selges samlet ca. kr 2200 (NOK)
Per Ole Klemetsrud
Irisveien 9
0870 Oslo 8
NORGE
Tlf.: 009-47-2-23 70 75 (Danmark)
Tlf.: (02) 23 70 75 (Norge)

KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP KLIP

MSX BRUGER-KLUBBEN.
KASSERER PREBEN LUND
HYLDESPJÆLDET
TØMMERSTRÆDET 19
2620 ALBERTSLUND

**** SVI BIB ****




SVI BIB

SVI BIB er KUN til SVI 318-328

SVIBIBS's POSTGIRONR. er:
4077385

Bestilling af programmer gøres på postgiro hvorpå der skrives hvilke programmer du ønsker,samt om du vil have disk eller bånd.
Husk også at skrive din egen adr. tydeligt helst BLOKBOGSTAVER.

PRISER

Bånd	-	25kr.
5.25" Disk	-	25kr.
Pris pr. program	-	10kr.
Pris pr. programpakke	-	15kr.
Udlistning af program	-	5kr.

Programpakkerne er lavet på den måde at man loader alle programmerne ind på en gang , og kan så vælge ved hjælp af en meny hvilket program man ønsker at køre derefter kan man stoppe og vende tilbage til menyen ovs.ovs.

HUSK når I bestiller programpakker at skrive hvilken katarogi I ønsker det pågældende nr. fra.

Programmerne bestilles hos:

Per Underlien
Rytterhusene 44
2620 Albertslund
GIRONR. 4077385

Programmer der ønskes optaget i SVIBIB sker på samme adr.
Det er jo sådan at vi NU GIVER
2 bånd eller
5 5.25" disk
For de programmer der optages i SVI BIB.

Hilsen Per

Liste af programtitler i SVIBIB :

Nyttige programmer :

Database , Editor , Hushold ,Matte, Mc scroll , Ramdis , Rentereg , Rulleteks , Screen Rig , Sdump.mc, Sorter , Sprite ed , Sprog , Tegner , Talkback , Taperut , Tips Kalender , Supertegn , Periode

Spil :

Aladdin , Bovla , Bjerg , Darts , Diktator Franz , Hell , Skyd nu , Oil

Spil pakker :

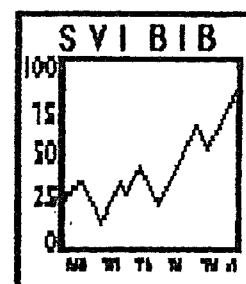
Nr. 1 : Male , Pyramide , Labyrant
Nr. 2 : Pusle , Attack , Roulette
Nr. 3 : Flight , Sur , War
Nr. 4 : Mons , Hopper , Star
Nr. 5 : Moon , Myre , Slange
Nr. 6 : Robot , Pengo , Snowball
Nr. 7 : Danmark , Frogger
Nr. 8 : Skyttepro , Lemona

Musik pakker :

Nr. 1 : Rock musik 9 numre
Nr. 2 : Rock musik 9 numre
Nr. 3 : Musik 9 numre
Nr. 4 : Lyd Kim , Sound Kim , Orgel

Andet pakker :

Nr. 1 : Rio , Lys , Ghost , Børge
Symtri , Grafik



Kun til disk :

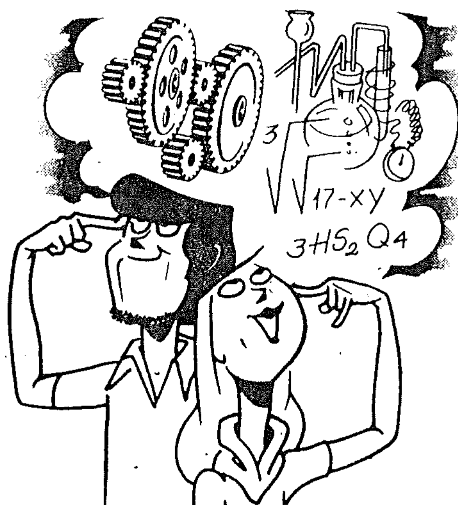
Budgetpro. , Dbase 1 no.1 , Grafen
Tegnepro. , Telefonpro.

****** BRØK REGNING AF KIM ******

BRØK REGNING AF KIM

Da jeg sad og rodede i mine gamle disketter, fandt jeg et program jeg havde lavet dengang jeg gik i folkeskolen og havde det skrækindjagende emne brøkregning. Da jeg efterhånden var blevet godt træt af at forkorte brøkker i hånden, lavede jeg dette program som kan gøre det for en. Man taster simpelt hen brøkken ind, venter et halvt sekund, og skriver den forkortede brøk ind i sin blæk-regning. Desuden har jeg lavet programmet som en regnemaskine med alle fire regnearter, så man heller ikke selv behøver at skulle sidde at gange med den omvendte når man skal dividere brøkker. Som en sidste detalje er der indbygget memory, hvis det er mere omfattende brøkregnestykker man skal i gang med.

Af: Kim Andersen




```
10 COLOR 15,1:CLS:WIDTH40
20 PRINT"          >>> BRØK <<<":PRINT
:PRINT" ENTER = FORKORTELSE AF BRØK
":PRINT" SPACE = NY BRØK":PRINT" M
    = MEMORY":PRINT" R      = RECALL
 MEMORY (skriv 'R'             ø
verst på brøkstregen)
40 ONERRORGOTO350
50 KEYOFF
60 GOSUB340:LOCATE1,11:LINEINPUTA$:
IFA$="r"ORA$="R"THEN180ELSEA=VAL(A$
):LOCATE1,13:LINEINPUTA$:B=VAL(A$)
70 IFA*B=0THEN60
80 ONTTGOTO190,200,210,220
90 A1=A:B1=B:A$=INPUT$(1):L=PEEK(63
073!)
100 IFA$=CHR$(32)THEN60
110 IF A$=CHR$(13)THEN230
120 IFA$="+"THENTT=1:GOTO60
130 IFA$="-"THENTT=2:GOTO60
140 IFA$="/"THENTT=3:GOTO60
150 IFA$="*"THENTT=4:GOTO60
160 IF A$="M"ORA$="m"THENA3=A:B3=B
170 GOTO90
180 IFA3*B3=0THEN50ELSEA=A3:B=B3:GO
TO240
190 A=B*A1+A*B1:B=B*B1:GOTO230
200 A=B*A1-A*B1:B=B*B1:GOTO230
210 SWAPA,B
220 A=A*A1:B=B*B1
230 X=1
240 LOCATE0,0:IFA>BTHENY=A:Y1=BELSE
Y=B:Y1=A
250 R=Y-(INT(Y/Y1)*Y1)
260 IFR>0THENY=Y1:Y1=R:GOTO250
270 A=A/Y1:B=B/Y1
280 A2=A
290 IFA=>BTHENC=INT(A/B):A2=A-B*CEL
SEC=0
300 IFA2=0THENF=1ELSEF=0
310 GOSUB340:IFC>0THENLOCATE0,12:PR
INTC
320 T=LEN(STR$(C))*SGN(C):IFF=0THEN
LOCATET,11:PRINTA2:LOCATET,13:PRINT
B
330 TT=0:GOTO90
340 LOCATE0,11:PRINTCHR$(27)"J":LOC
ATE1,12:PRINT"WWWWWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWW":RETURN
350 LOCATE11,8:PRINT">>> ERROR <<<"
;
360 A$=INPUT$(1):PRINTCHR$(27)"1"
370 RESUME60
```

OBS! 'W' i linie 340 skal være grafiktegnet på GRAPH + '-'

MSXDOS, CP/M OG MS-DOS

Kurt giver dig her forklaring hvad et disk-operativ-system er og hvordan du kan bruge det. For alle computere med disk.

Når du tænder en SVI328 eller MSX vil den starte op i Basic. Indsætter du lige efter du har tændt computeren en CP/M eller MSXDOS system diskette vil computeren ikke bruge Basic, men i stedet indlæse CP/M eller MSXDOS operativ systemet. På en PC vil man altid indsætte en system diskette, for at computeren kan starte operativ systemet MS-DOS. Et operativ system, CP/M, MSXDOS eller MS-DOS, er ikke en anden udgave af et Basic sprog. I et operativ system er det ikke muligt direkte at skrive programmer som man kan gøre i Basic. Operativ systemerne er specielt beregnet til at styre disk, ændre filer og køre færdige programmer fra disk. Der er ikke særligt mange kommandoer og langt de fleste bruger diskette på en eller anden måde. Det er derfor naturligt at kalde dem disk operativ systemer eller blot DOS. De tre DOS'er er forskellige men er opbygget efter samme principper, og de har derfor mange lighedspunkter.

I DOS'en kan du altså ikke skrive programmer, men du kan køre færdige programmer - beregnet til DOS'en, ved blot at skrive deres navn og trykke Enter. For at kunne skrive programmer som kan bruges i DOS'en, skal du have et programmerings sprog, som startes fra DOS'en. Dette kunne f.eks. være en Pascal (f.eks. Turbo Pascal) eller Assembler, hvor du skriver et program som bliver oversat til maskinkode. Det oversatte program kan derefter køres fra DOS'en. Derimod kan Basic programmer ikke køres fra DOS'en da de ikke er oversat til maskinkode.
Bemærk: I blad nr. 3 fra i år viste vi Basic programmer til at definere funktionstaster i de tre DOS'er: CP/M, MSXDOS og MS-DOS. Her var det ikke selve Basic programmet som blev kørt fra DOS'en, derimod oprettede Basic programmet et andet program som kan bruges i DOS'en.

Kommandoerne i MSXDOS og MS-DOS er næsten helt ens, der er blot flere i MS-DOS. Microsoft har skrevet MS-DOS som operativ system til PC'ere, der jo er 16 bit computere. Da Microsoft lavede MSX standarden opfandt de en 8 bit udgave af operativ systemet som fik navnet MSXDOS. Da PC'ere har 16 bit processorer (8088) og MSX har 8 bit (Z80) kan MS-DOS programmer ikke bruges på MSX og omvendt - derimod er der andre ligheds punkter. Som jeg har forklaret tidligere i andre numre af bladet er disk-formatet, måden programmer og data gemmes på disketten, ens. Dette giver mulighed for at udveksle taldata fra regneark, breve og andre tekstbehandlings tekster mellem MSX og PC. Det største problem er som regel at MSX bruger 3.5" disketter hvorimod PC'ere almindeligvis bruger 5.25" disketter. Dette kan klares ved at tilslutte et 5.25" diskdrev som ekstra drev på MSX'en.

CP/M er lavet af det amerikanske firma Digital Research, nogle år før Microsoft lavede MS-DOS, som operativ system for Intels 8080 8 bits processoren. Z80 som bruges i SVI328 og MSX er en forbedret udgave af 8080 og kan alt hvad 8080 kan plus meget mere.
I CP/M kan du gøre nogenlunde de samme ting som i de to andre DOS'er - kommandoerne har blot andre navne. Internt har Microsoft opbygget MSXDOS ligesom CP/M og dermed kan CP/M og MSXDOS køre de samme programmer. Det er dog muligt at lave specielle programmer som kun kan fungere i CP/M eller kun i MSXDOS, men alle de kendte programmer så som Wordstar, Dbase II og Multiplan fungerer glimrende.

**** MSXDOS, CP/M OG MS-DOS ****

I MSXDOS og CP/M er der ikke så mange kommandoer men de der er tilgengæld nyttige. MS-DOS har flere kommandoer, som jeg ikke her vil komme specielt ind på. Kommandoerne i DOS'erne er næsten alle beregnet til at styre disk og ændre filer på disketten.

Den mest brugte kommando er DIR som viser filerne på disketten. DIR er en forkortelse af directory (katalog). Skriver du blot DIR og trykker Enter vil du få en liste med en fil på hver linje. Her kan du se filnavnene, deres filtype, størrelse og dato (kun i MSXDOS og MS-DOS). Nederst vil der stå hvor meget fri plads der er på disketten (kun i MSXDOS og MS-DOS).
Filnavnet og filtypen er de to dele af filens fuldstændige navn, de er adskildt af et punktum. Filnavnet kan være på op til 8 bogstaver og filtypen op til 3 bogstaver. Der kan dog også indgå tal og almindelige tegn i filnavnet. Størrelsen angiver hvor mange bytes filen fylder. Datoen bruges som regel ikke i MSXDOS da de fleste MSXere ikke har indbygget et seperat ur.
Efter DIR kan du angive en fil-specifikation. Dette giver mulighed for kun at se den del af filerne som svarer til specifikationen. Skriv:
DIR DUCKY
Viser filerne med navnet DUCKY og vilkårlig filtype. I fil-specifikationen kan * og ? bruges som specielle tegn ("wildcards") til at gøre fil-specifikationen mere generel. ? kan stå et vilkårligt tegn og * betyder et eller flere vilkårlige tegn.
Skriver du: DIR IN?TEKST.DOC
vil du se alle de filer Hvis navne begynder med IN, har et vilkårligt tegn, slutter med TEKST samt har filtypen DOC. Følgende filnavne, samt mange andre, vil altså kunne bruges:
IN2TEKST.DOC , IN9TEKST.DOC , INPTEKST.DOC
* bruges egentlig oftere. Den angiver at resten kan være vilkårlige tegn. DIR IN*.DOC kunne altså blandt andet give dig:
IN2TEKST.DOC , INPTEKST.DOC , IN9TEXT.DOC , INSTALL.DOC , INXS.DOC
Du kan også sætte * og ? i filtypen for at angive vilkårlige filtyper. ??? er det samme som en * og betyder vilkårlig filtype. *.* betyder alle filnavn med alle filtyper, altså alle filer på disketten.
DIR K*.* kunne give:
KONVERT.COM , KEY.TXT , KURT.DOC , KURT.COM , KLODS.BAS

I fil-specifikationen kan også indgå navnet for forskellige diskdrev. Efter bogstavet for diskdrevet skal der være et kolon. DIR B: viser altså kataloget for et ekstra diskdrev, der har navnet B. Efter B: kan der angives alle de samme fil-specifikationer som forklaret ovenfor.

MSXDOS og MS-DOS: Har du kun et diskdrev vil drev B være det samme A drevet, men computeren har en intern logik til at lade som om den har to diskdrevs. Skriver du derfor DIR B: selvom der kun er tilsluttet et diskdrev, vil computeren bede dig om at indsætte disketten som skulle sidde i drev B. Den viser dig nu kataloget for denne nye diskette, og husker at det er B drevets diskette som sidder i diskdrevet. Skriver du derfor DIR som egentlig betyder DIR A: vil den bede dig om at indsætte disketten for A drevet før den viser kataloget. Bemærk at computeren ikke "kan se" om du allerede har skiftet disketten inden en kommando bruges. Dette princip med "logiske drev" er meget smart, men du skal nogle steder passe på at du ikke kommer til indsætte en forkert diskette. Mere om logiske drev senere.

MSXDOS og MS-DOS: DIR kommandoen har to ekstra muligheder. Hvis du ikke er interesseret i den enkeltes fils størrelse kan du få en mere kompakt liste med flere filnavne pr. linje ved at skrive DIR/W. Hvis du har 40 tegn på skærmen får du 2

**** MSXDOS, CP/M OG MS-DOS ****

filnavne/linje har 80 tegn får du 5. W står for Wide, og kan sagtens benyttes med alle mulige fil-specifikationer, f.eks.: DIR B:/W , DIR/W B:KURT*.*
Det er lige godt om /W sættes før eller efter fil-specifikationen. Jeg synes selv at DIR/W er mere andvendelig end blot DIR fordi den kan give en meget større oversigt på skærmen.
En anden mulighed er at skrive /P, som kan benyttes hvis kataloget er for stort til at være på skærmen. /P skriver højst en side (Page) ad gangen, venter til du trykker på en tast før næste side kommer. /P kan ligesom /W benyttes sammen med diverse fil-specifikationer og det er også muligt at benytte begge i samme DIR kommando. En anden mulighed for at stoppe en lang udskrift er at trykke CTRL S som vil pause udskriften indtil du trykker på en anden tast. CTRL S kan bruges mange andre steder end ved kataloger, hvor udskriften er meget lang.

CP/M: Som forklaret giver DIR kommandoen i CP/M ikke andre oplysninger og filernes størrelse. For at se dette kan du bruge STAT programmet. Efter STAT angiver du en fil-specifikation, for at få hele disketten skriver du altså:
STAT *.*
Bemærk: Når du sætter en ny diskette i et diskdrev skal trykke Ctrl og C, for at få CP/M'en til at acceptere den nye diskette. Gør du det ikke vil CP/M'en stadig bruge visse oplysninger fra den gamle diskette, som ikke har noget at gøre med den nye. Bliver du i et program bedt om at skifte diskette, skal du ikke trykke Ctrl C, her er det op til programmet selv at initialisere den nye diskette.

Før en ny diskette kan tages i brug skal den formateres (klargøres). Dette gøres med kommandoen FORMAT. Formateringen opdeler disketten i spor og sektorer. Inden dette kan disketten ikke bruges til at gemme programmer og data på. I MS-DOS skal du lige efter FORMAT skrive det diskdrev hvor disketten skal formateres, f.eks. B:. I CP/M og MSXDOS spørger FORMAT programmet/kommandoen om hvilket diskdrev der skal formateres.
Pas på du ikke kommer til at formatere en diskette som indeholder vigtige programmer. Efter FORMAT vil alt være slettet.
Når du har formateret en diskette kan du kopiere DOS systemet over på den ny diskette. Dette giver dig mulighed for at starte DOS'en op med disketten.
MS-DOS: Her er det egentlig nemmest hvis du gør det samtidig med formateringen, ved at skrive:
FORMAT B:/S
Dette vil formatere disketten i drev B, og lægge system filerne over. System filerne er: IO.SYS, MSDOS.SYS og COMMAND.COM
Filerne IO.SYS og MSDOS.SYS er skjulte filer som ikke kan ses med DIR.
MSXDOS: Efter formateringen kopierer du MSXDOS.SYS og COMMAND.COM over på den nye diskette ved at bruge kommandoen COPY.
CP/M: Her bruger du programmet SYSGEN, som henter CP/M systemet fra de specielle system spor på en diskette og lægger dem over på din nye diskette. System sporene på disketten kan ikke bruges til andet end systemet, så CP/M systemet optager ingen plads som kan bruges til andet.

Næst efter DIR er kopiering nok den mest benyttede kommando fordi den giver dig mulighed for at flytte dine programmer og data fra diskette til diskette.

COPY i MSXDOS og MS-DOS har mange varianter; det simpelte er nok kopiering af en fil fra en diskette til en anden. For at kopiere filen VIDEO2.BAS, skriv da:

**** MSXDOS, CP/M OG MS-DOS ****

COPY A:VIDEO2.PIC B:
Har du to diskdrev vil filen VIDEO2.BAS øjeblikkelig blive kopieret over på B drevet.
Har du kun et diskdrev vil computeren udnytte B som "logisk drev", ved at indlæse filen i hukommelsen og bede dig indsætte en diskette for drev B. Hvis filen er meget stor eller du har angivet fil-specifikation i stedet for et filnavn, kan hele kopieringen måske ikke foregå på et skift fordi det hele ikke kan være i hukommelsen på en gang. Computeren vil derfor flere gange bede dig om at skifte mellem diskette for drev A og drev B. Det er på disse tidspunkter at man alvorlig overvejer at anskaffe sig et ekstra diskdrev.
Det er meget vigtigt at du under disse kopierings skift ikke indsætter forkert diskette, det kan lave kuk i disketten hvis computeren får mulighed for at skrive på en forkert diskette. Alle programmer og data kan blive ubrugelige, pas derfor på. Det kan være en god ide at skrive beskytte disketten der skal kopieres fra (source disk), inden kopieringen startes. Dette sikrer at du ikke kan komme til at ødelægge source-disketten. Du skal selvfølgelig ikke skrive beskytte den diskette der skal kopieres til (destination disk).
Bemærk at du ved kopiering, også med et diskdrev, foretager en kopiering fra A til B, du skal ikke skrive:
COPY DUCKY.ASM A:
Computeren vil her tro at du vil kopiere fra drev A til samme drev. Da du ikke har angivet et nyt filnavn efter A: kommer der en fejlmeddelelse fordi det ikke er muligt at kopiere en fil oveni sig selv. Havde du angivet et nyt filnavn efter A: f.eks.:
COPY DUCKY.ASM A:DUCKY.BAK
Dette giver ingen problemer, her vil indholdet af filen DUCKY.ASM blive kopieret over i en fil med navnet DUCKY.BAK på den samme diskette, så de vil være identiske efter kopieringen. En sådan navneændring kan også benyttes ved kopiering mellem to diskdrev.

I COPY kommandoen kan du bruge * og ? til at lave forskellige filspecikationer på samme måde som det foregik ved DIR kommandoen. Vil du kopiere en hel diskette kan dette gøres med:
COPY A:*.* B:
Hele disketten i drev A vil blive kopieret over på disketten i drev B. Det er selvfølgelig også muligt at kopiere den "anden vej":
COPY B:*.COM A:
Alle filer med filtypen COM på diskdrev B, vil blive kopieret over til drev A.

Det er ikke nødvendigt både A: og B:. Står der A> er dit aktuelle diskdrev A: og computeren bruger altid dette diskdrev når andet diskdrev ikke er angivet. Hvis der står A> vil:
COPY FLEMMING.PAS B:
vil filen FLEMMING.PAS blive kopieret over på diskdrev B. Derimod:
COPY B:KURT3.GEN
Dette er nok filen fra diskdrev B til drev A. Der er ikke angivet nogen destinations drev så computeren bruger det aktuelle drev, altså A.
Ønsker du at bruge B diskdrevet som aktuelt drev, kan du skifte ved at skrive B: , og du kan selvfølgelig senere skifte tilbage til A ved at skrive A:

CP/M: Til kopiering i CP/M bruges programmerne FILECOPY eller PIP. PIP kan benyttes til kopiering hvis man har to diskdrev, benyt ellers FILECOPY. I CP/M er der ikke mulighed for "logiske drev" som der er i MSXDOS og MS-DOS. For at kopiere filen VEKTOR.COM fra disk A til B, skrives:
PIP B:=A:VEKTOR.COM
Bemærk at der skrives B = filen fra A. Det er selvfølgelig også muligt at bruge * og ? i filnavnene til at angive flere filer.

**** MSXDOS, CP/M OG MS-DOS ****

Da PIP ikke er en kommando men et program, skal dette program ligge på den ene af disketterne, før det kan bruges. Skal du foretage meget kopiering kan PIP indlæses en gang ved blot at skrive PIP. Herefter kan du så skrive de forskellige fil specifikationer til kopieringen. Når du er færdig kan du forlade PIP ved at trykke Ctrl C.
FILECOPY er ikke særlig svær at bruge, skriv blot FILECOPY. FILECOPY programmet vil blive indlæst og vil spørge dig hvad du ønsker at kopiere. Kopierer du store filer eller flere filer på en gang kan det være nødvendigt med adskillige diskette skift.

Til at slette filer benyttes i MSXDOS og i MS-DOS DEL kommandoen. I CP/M hedder den tilsvarende kommando ERA. Efter kommandoen angiver du et filnavn som skal slettes. F.eks.:
MSX-DOS/MS-DOS: DEL KURT.BAS
CP/M: ERA KURT.BAS
Det er selvfølgelig også muligt at bruge * og ? til at angive at flere filer skal slettes på en gang. DEL *.* og ERA *.* sletter en hel diskette.
Har du skrive beskyttet disketten med tape (5.25" disk) eller med den lille tap (3.5" disk) kan disketten ikke slettes før skrive beskyttelsen er blevet fjernet. Tag disketten ud og gør dette inden sletning, kopiering eller anden brug hvor der skrives på disketten.

Vil du ændre navn på en fil kan dette gøres med REN (rename). I MSXDOS og MS-DOS skriver du:
REN TEXTSORT.PAS QSORT.PAS
Dette vil ændre navnet på filen TEXTSORT.PAS, til navnet QSORT.PAS.
I MSXDOS og MS-DOS skrives det gamle navn først og det nye navn bagefter.
I CP/M er det lige omvendt. Her skrives nyt navn = gammelt navn, altså sådan:
REN QSORT.PAS=TEXTSORT.PAS
Det er selvfølgelig ikke muligt skifte et filnavn til et nyt som en anden fil har i forvejen.

Filtypen som er angivet efter selve filnavnet kan egentlig være hvad som helst, men nogle ord er fast defineret. Typen COM er til programmer som er skrevet til at køre i DOS. Programmerne kan køres ved blot at skrive selve filnavnet uden filtypen COM. I MSDOS findes der også filtypen EXE som også er programmer, om det hedder COM eller EXE afhænger af programmets størrelse - men det giver ingen forskel i brugen, skriv blot filnavnet for at starte programmet.
BAT (MSDOS/MSXDOS) og SUB (CP/M) er "batch" filer. Dette er filer som indeholder en liste med kommandoer som du ellers skulle taste en efter en på tastaturet. Dette giver mulighed for med en kommando at udføre mange kommandoer.
Resterende filtyper kan så vælges frit men det er en god ide at holde visse standarder. Brug filtypen BAS til Basic programmer, så er de nemmere at finde i DIR katalogerne, bemærk på PC bruges denne filtype i Basic hvis intet andet angives.

En tekstfil er som navnet siger fil indeholdende tekst. De tekster man skriver i en tekstbehandling eller editor vil ende som en tekstfil. Når man skriver et Pascal eller Assemblerprogram, vil man bruge en editor til at skrive selve teksten, som Pascal-kompileren eller assembleren derefter vil oversætte til det færdige program. Hvis kompileren og editoren er bygget sammen i et program, som det er tilfældet med f.eks. Turbo-Pascal, er det ikke nødvendigt først at gemme teksten i en tekstfil inden den oversættes.
En tekstfils opbygning er egentlig simpel i opbygningen. Den indeholder teksten, med linjeskifts koderne: 13 (enter) og 10 (linefeed) i slutningen af hver linje. For at markere slutningen af filen skal der

**** MSXDOS, CP/M OG MS-DOS ****

være en kode 26 (eof), som er ASCII-koden for End Of File.
I alle DOS'er vil tekstfilens indhold blive vist på skærmen med kommandoen TYPE. Til forskel fra f.eks. DIR skal filnavnet angives præcist, man kan ikke bruge * og ? til at se flere filer.
Hvis der kommer en masse vrøvl ud på skærmen når du TYPE'r en fil er det fordi det ikke er en tekstfil. Dette vil der komme hvis TYPE'r en fil med filtypen COM.
Ofte vil programmer som ikke har indbyggede hjælpe funktioner, være ledsaget af en hjælpe-tekstfil som forklarer programmets muligheder og funktioner. Navnet på en sådan fil er normalt programnavnet efterfulgt af filtypen DOC, HLP eller TXT. En sådan fil kan gennemlæses ved at bruge TYPE kommandoen. Hvis den er meget lang kan den pauses med Ctrl S, eller helt afbrydes med Ctrl C.
Hvis du TYPE'r en fil fra Wordstar tekstbehandlingen bliver udskriften mærkelig, men det er dog muligt at "forstå" teksten. Dette skyldes at Wordstar har lagt forskellige styrekoder ind i teksten, ved at lægge 128 (bit 7 sat) til nogle at tegnkoderne. De linjeskift Wordstar selv har placeret er derfor blevet til 128 + 13 = tegnkode 141.
Bruger du Wordstar's N-kommando i stedet for D, vil Wordstar virke som en editor og ikke sætte specielle styrekoder. Her som i andre editorer vil det ikke være muligt at udnytte tekstbehandlings faciliteter så som lige højremargen.

Vil du have en tekstfil skrevet ud på printer kan det alt efter DOS type gøres på forskellig måde. I både CP/M, MSXDOS og MS-DOS kan det gøres ved at bruge Ctrl P. Skal filen LOOK.DOC udskrives:
TYPE LOOK.DOC
inden du trykker på enter skal du trykke på Ctrl P, og derefter enter. Filen vil nu blive udskrevet på både skærm og printer. Lige så snart computeren melder klar skal du trykke på Ctrl P igen hvis du bruger CP/M. Bruger du MSXDOS eller MS-DOS skal du trykke Ctrl N.
Gør du ikke dette vil alt hvad du herefter fortager i DOS'en også blive udskrevet på printeren, en såkaldt printer-spool.
I MSXDOS og MS-DOS kan du også kopiere filen ud på printeren. Skriv:
COPY LOOK.DOC PRN
PRN (og LST) er et "device" navn for printeren, ligesom A: og B: er det for diskdrevne, bemærk PRN (og LST) skrives uden kolon efter.
I MS-DOS kan en tekstfil også udskrives med kommandoen PRINT, som er et mindre program der udskriver filer i "baggrunden". At udskriften foregår i baggrunden betyder at udskriften kan foregå mens du bruger computeren til andre opgaver.

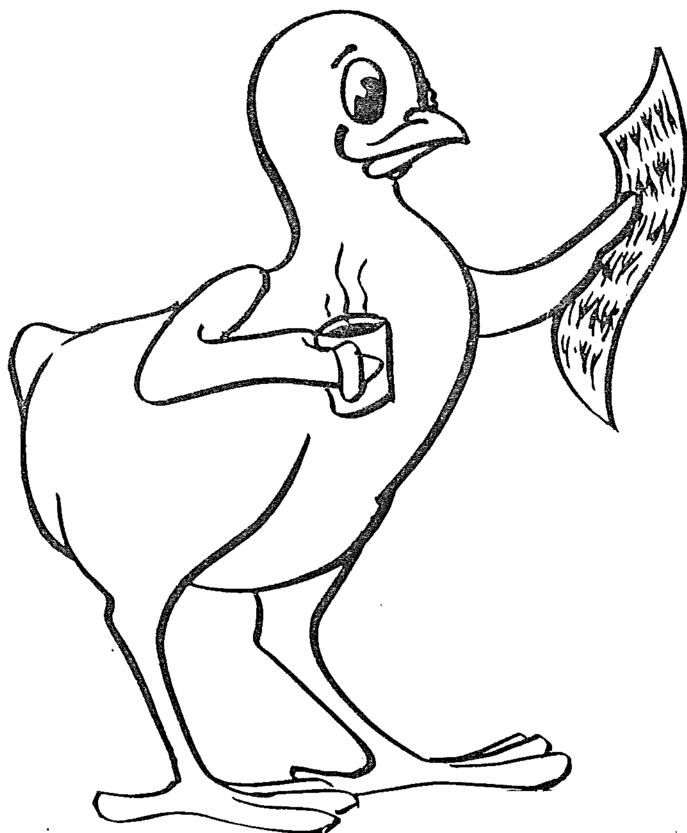
I MSXDOS kan du "gå op" i MSX Basic med kommandoen BASIC. Efter BASIC kan skrive et filnavn på et Basic program som skal indlæses og køres samtidig med Basic startes. For at gå tilbage til MSXDOS med kommandoen CALL SYSTEM, i stedet for CALL kan du skrive blot en _ (underscore - understregning). Der skal sidde en MSXDOS system diskette i diskdrevet.
I GWBasic på PC kan du ligeledes gå ud i MS-DOS med kommandoen SYSTEM, bemærk at der ikke skrives CALL foran.

Hvis du ønsker at skrive programmer som kan benyttes i DOS vil jeg anbefale dig at bruge Pascal. Pascal er ikke lige at gå til, men det er et sprog som giver dig mulighed for at lave store og kraftige programmer. Den bedste Pascal du kan få er nok Turbo Pascal fra det amerikanske firma Borland. Borland er meget billige - ihvertilfælde når man ser på kvaliteten - du får meget for pengene. Turbo Pascal version 4 til PC koster i Danmark kr. 890.- (+ 22% moms). Til CP/M (og dermed MSXDOS) kan du få en version 3 til blot kr. 560.- (+ 22% moms). Du kan kontakte Borland afdelingen hos

**** MSXDOS, CP/M OG MS-DOS * TIPS ****

Polydata, som er den danske importør, på telefon 02 27 74 00. Hvis du skal have CP/M versionen til din computer, skal du oplyse hvilket disk-format den skal have. Dette afhænger af dit diskdrev: Enkelt/dobbelt sidet, 40/80 spor og antal sektorer pr. spor. Skal du bruge CP/M versionen til MSXDOS kan du sige at formatet er det samme som 3.5" PC (blot enkel sidet). Skulle du få problemer kan du jo give dem en formateret diskette - så må de på den måde selv kunne finde ud af formatet.

Your Disk-Jokey Kurt.



TIPS

Af JOHNNY B MUSIK

Til båndfolket en lille rutine du kan merge med dit program hvis du ønsker at det skal loade og runne et nyt Basic program fra bånd indtast denne linie og gem den som aschii fil =csave-navn-,a

```
60000 Z$="RUN"+CHR$(13):FORQ=-1040
TO-1000:POKEQ,0:NEXT:POKE-3078,240
:POKE-3077,251:POKE-3076,244:POKE-
3075,251:FORQ=1TOLEN(Z$):POKEQ-104
1,ASC(MID$(Z$,Q,1)):NEXT:CLOAD
```

Endnu en lille merge linie til genopretning af F taster uden at slukke computeren

```
60000 DEFUSR=&H3E:A=USR(0):KEYON
```

Denne linie sletter de F taster du eller et program har lavet og kommer op med de samme som ved opstart af computeren

OBS: Du kan også skrive linien direkte i basic på skærmen uden linienummer og det samme sker

For Disk folket her en repeat

POKE &hf346,1:_SYSTEM
booter Dos direkte fra Basic

Endvidere
POKE &Hf247,1 gør B drive
til default drive

Til MSX II folket dette

VDP(10)=0
giver 60 hrz interrupt, godt til spil de kører hurtigere

OBS til jer der sidder derhjemme kom dog ud af busken med jeres Tricks,Pokes og Tips og lad vi andre få glæde deraf.

HEJ JOHNNY.

SOFT MED POST

Code Masters

Vampire - arc.adv. Brok the Brave, må finde og bekæmpe grev Dracula i hans slot med 95 rum 38.-

Pro-Brooker - billiard for 1/2 spil. 38.-

Bmx Simulator - [illegible] gennem [illegible] din modstander i [illegible] inden han [illegible] det med dig. 1/2 sp. [illegible]

[illegible]

[illegible] - en [illegible] 4[illegible].-

[illegible]

[illegible]

Ultimate

[illegible] grafik 4[illegible].-

[illegible] - Et rumskib fra Markon er blevet angrebet af de barbariske [illegible] [illegible] The [illegible] 3D scroll, graf. [illegible]

[illegible]

[illegible]

Firebird

Elite - [illegible] rumskib [illegible] på en fantastisk [illegible] nye planeter. Bekæmp pirater som vil [illegible] som [illegible] af [illegible]; 3D vektorgrafik arc.str., m. fyldig baggrunds historie og instruktion - bånd 215.-

Elite - disk 249.-

Konami

F-1 Spirit - vejen til Formula-1, nyt race-spil m. Stock-car, Rally, F-3, Endurance, F-3000 og tilslut Formula-1 hvis du kan kvalificere dig; 1/2 spillere; indeholder ekstra lydchip 277.-

Nemesis II - fortsættelsen til Konamis flotte rumspil, kampen om planeten Nemesis, 128K, ekstra lydchip 277.-

HAL

Dunkshot - Basketball 215.-

Eggerland 2 - 100 labyrint baner 277.-

Anglosoft

Print X'press - Lav plakater, postkort, brevpapir og design af udskrifter m. figurer og 19 forskellige tegnsæt; [illegible] og [illegible] kan [illegible] - bånd og disk [illegible]

[illegible]

[illegible] Code Tutor & Monitor - [illegible] maskinkodens [illegible] og [illegible] til maskinkode begyndere [illegible]

Hi-soft

Devpac - bedste assemblersystem [illegible]

Devpac 80 - disk [illegible]

Kuma bøger

Behind the Screens of the MSX 1[illegible].-

Programming in MSX Basic [illegible]

[illegible] for MSX [illegible]

The MSX Red Book [illegible]

[illegible] disk, dobb.sidet 10 stk. [illegible]

[illegible] disk, dobb. sidet [illegible] stk. [illegible]

[illegible] er inkl. dansk moms. Der tages forbehold for trykfejl og prisændringer. [illegible] forsendelse ved forudbetaling med check/giro. Bestillinger under kr. [illegible].- tillægges dog kr. 20.- ekspeditionsgebyr.

Ring 01 [illegible] mellem 16-20 mandag til lørdag, eller send en check eller en giro på gironr. 345 96 32 med dine bestillinger. Oplys evt. telefonnr.

Salg til hele Skandinavien. Ved køb fra Sverige/Norge indbetal den danske pris på et postgiro-kort, og skriv hvad du ønsker på bagsiden.

D a t a c r a f t
N.J. Fjords alle 20
DK-1957 Frederiksberg C
tlf. 01 393994